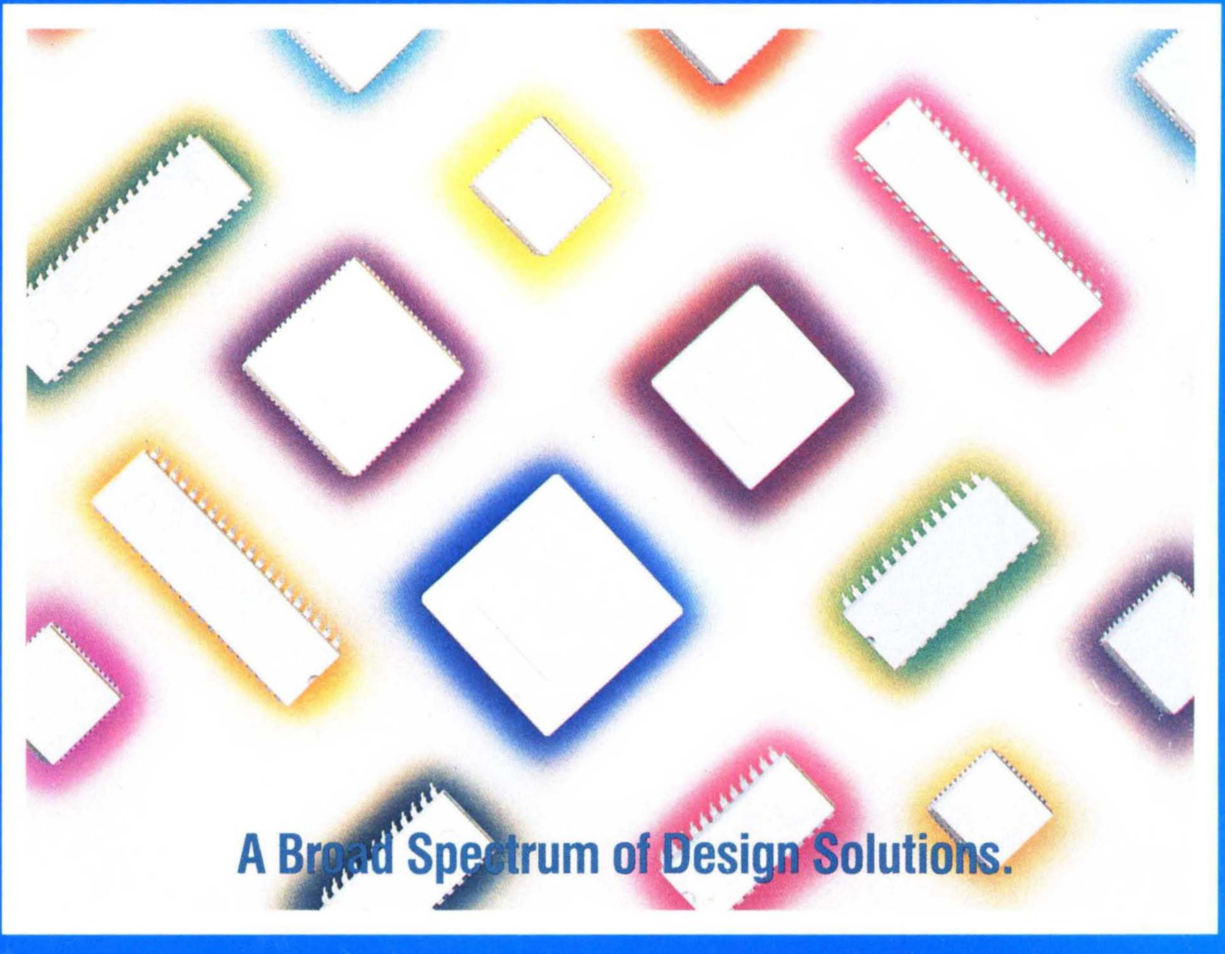
DL139

MOTOROLA



A Broad Spectrum of Design Solutions.

MICROPROCESSOR, MICROCONTROLLER AND PERIPHERAL DATA

VOLUME II

Motorola's Microcontroller and Microprocessor Families 1
Volume I

Reliability 2
Volume I

Data Sheets 3
Volume I and II

Mechanical Data 4
Volume II

Evaluation Modules 5
Volume II

Ordering Information Forms 6
Volume II

1 **Motorola's Microcontroller and Microprocessor Families**
Volume I

2 **Reliability**
Volume I

3 **Data Sheets**
Volume I and II

4 **Mechanical Data**
Volume II

5 **Evaluation Modules**
Volume II

6 **Ordering Information Forms**
Volume II

MOTOROLA

MICROPROCESSOR DATA
VOLUME II

Prepared by
Microprocessor Products Group

This book is intended to provide the design engineer with the technical data needed to completely and successfully design a microcomputer-based system. The Technical Summary and Advance Information data sheets for Motorola's microcontroller, microprocessor, and peripheral components are included.

The information in this book has been carefully checked; no responsibility, however, is assumed for any inaccuracies. Furthermore, this information does not convey to the purchase of microelectronic devices any license under the patent rights of the manufacturer.

Additional information on Motorola's new products and system development products are also included. For further marketing and application information, please contact:

Motorola Inc.
Microprocessor Products Group
Microcontroller Division
6501 William Cannon Drive West
Austin, Texas 78735-8598

Applications	512-891-2034
EVB/EVM/Development Systems	512-891-2034
Literature — Literature Distribution Center	602-994-6561
Marketing Information — Pricing	Local Sales Office
Marketing Information — Availability	512-891-2990

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not authorized for use as components in life support devices or systems intended for surgical implant into the body or intended to support or sustain life. Buyer agrees to notify Motorola of any such intended end use whereupon Motorola shall determine availability and suitability of its product or products for the use intended. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Employment Opportunity/Affirmative Action Employer.

Series A
Second Printing
©MOTOROLA INC., 1988
"All Rights Reserved"

Printed in U.S.A.

DATA CLASSIFICATION

Product Preview

Data sheets herein contain information on a product under development. Motorola reserves the right to change or discontinue these products without notice.

Technical Summary

Data sheets herein contain information on new products. Specifications and information are subject to change without notice.

Advanced Information

Data sheets herein contain information on new products. Specifications and information are subject to change without notice.

EXORciser is a register trademark of Motorola, Inc
HDS-300, MDOS, and BUFFALO are trademarks of Motorola, Inc
MS-DOS is a trademark of Microsoft, Inc
IBM is a register trademark of International Business Machines, Inc

TABLE OF CONTENTS

The Microprocessor/Microcontroller and Peripheral Data book consist of a two volume set. Refer to the table of contents and the master index for division of the chapters and locations of devices.

Page

Chapter 1 — Motorola's Microprocessor and Microcontroller Families

M68HC11/M6801/M6805/M6804 Families .... Vol. I
Development Support .... Vol. I
Single-Chip MCU Selector Guides .... Vol. I

Chapter 2 — Reliability and Quality Summary

Introduction .... Vol. I
Quality and Reliability System .... Vol. I
Packaging System .... Vol. I
Results and Conclusion .... Vol. I
Failure Rate Calculations .... Vol. I

Chapter 3 — Data Sheets

(See master Index for sequence)

Chapter 4 — Mechanical Data

Introduction 4-1
Package Dimensions 4-3

Chapter 5 — Evaluation Modules

Development Station 5-3
Low Cost MCU Evaluation Modules 5-5

Chapter 6 — Product Ordering Forms

MASTER INDEX

Device Number	Description	Page Number
MC2672	Programmable Video Timer Controller	Vol. I
MC2674	Advanced Video Display Controller	Vol. I
MC6800	8-Bit Microprocessor Unit	Vol. I
MC6801/ MC6803	8-Bit Microcontroller Unit	Vol. I
MC6801U4/ MC6803U4	8-Bit Microcontroller Unit	Vol. I
MC68701	8-Bit Microcontroller Unit	Vol. I
MC68701U4	8-Bit Microcontroller Unit	Vol. I
MC6802	8-Bit Microprocessor with Clock and RAM	Vol. I
MC6804J1	8-Bit Microcontroller Unit	Vol. I
MC6804J2	8-Bit Microcontroller Unit	Vol. I
MC6804P2	8-Bit Microcontroller Unit	Vol. I
MC68704P2	8-Bit Microcontroller Unit with EPROM	Vol. I
MC68HC04J2	8-Bit HCMOS Microcontroller Unit	Vol. I
MC68HC04J3	8-Bit HCMOS Microcontroller Unit	Vol. I
MC68HC04P4	8-Bit HCMOS Microcontroller Unit	Vol. I
MC68HC704P4	8-Bit HCMOS Microcontroller Unit with OTPROM or EPROM	Vol. I
MC6805P2	8-Bit Microcontroller Unit with 1K ROM	Vol. I
MC6805P6	8-Bit Microcontroller Unit	Vol. I
MC6805R2	8-Bit Microcontroller Unit with A/D Converter	Vol. I
MC6805R3	8-Bit Microcontroller Unit with A/D Converter	Vol. I
MC6805S2	8-Bit Microcontroller Unit with A/D Converter, SPI, and Three Timers	Vol. I
MC6805S3	8-Bit Microcontroller Unit with A/D Converter, SPI, and Three Timers	Vol. I
MC6805U2	8-Bit Microcontroller Unit	Vol. I
MC6805U3	8-Bit Microcontroller Unit	Vol. I
MC68705P3	8-Bit Microcontroller Unit with EPROM	Vol. I
MC68705P5	8-Bit Microcontroller Unit with EPROM	Vol. I
MC68705R3	8-Bit Microcontroller Unit with A/D Converter and EPROM	Vol. I
MC68705R5	8-Bit Microcontroller Unit with A/D Converter Secured EPROM	Vol. I
MC68705S3	8-Bit Microcontroller Unit with A/D Converter, SPI, EPROM, and Three Timers	Vol. I
MC68705U3	8-Bit Microcontroller Unit with EPROM	Vol. I
MC68705U5	8-Bit Microcontroller Unit with Secured EPROM	Vol. I
MC68HC05A6	8-Bit Microcontroller Unit	Vol. I
MC68HC05B4	8-Bit Microcontroller Unit	Vol. I
MC68HC05B6	8-Bit Microcontroller Unit	Vol. I
MC68HC05C2	8-Bit Microcontroller Unit	Vol. I

MASTER INDEX (Continued)

Device Number	Description	Page Number
MC68HC05C3	8-Bit Microcontroller Unit	Vol. I
MC68HC05C4	8-Bit Microcontroller Unit	Vol. I
MC68HC05C8	8-Bit Microcontroller Unit	Vol. I
MC68HC05C9	8-Bit Microcontroller Unit	3-939
MC68HC05L6	8-Bit Microcontroller Unit	3-941
MC68HC05M4	8-Bit Microcontroller Unit	3-983
MC68HC05P1	8-Bit Microcontroller Unit	3-985
MC68HCL05C4	8-Bit Microcontroller Unit	3-1012
MC68HCL05C8	8-Bit Microcontroller Unit	3-1053
MC68HSC05C4	8-Bit Microcontroller Unit	3-1092
MC68HSC05C8	8-Bit Microcontroller Unit	3-1133
MC68HC705B5	8-Bit Microcontroller Unit with OTPROM	3-1173
MC68HC705C4	8-Bit Microcontroller Unit with OTPROM or EPROM	3-1175
MC68HC705C8	8-Bit Microcontroller Unit with Standard EPROM	3-1177
MC68HC805B6	8-Bit Microcontroller Unit with EEPROM	3-1219
MC68HC805C4	8-Bit Microcontroller Unit with OTPROM or EEPROM	3-1260
MC146805E2	8-Bit Microcontroller Unit	3-1302
MC146805F2	8-Bit Microcontroller Unit	3-1326
MC146805G2	8-Bit Microcontroller Unit	3-1343
MC6809	8-Bit Microcontroller Unit	3-1363
MC6809E	8-Bit Microcontroller Unit	3-1397
MC6810	128×8 Bit Static RAM	3-1432
MC68HC11A0	8-Bit HCMOS Microcontroller Unit with SPI, SCI, and A/D Converter	3-1437
MC68HC11A1	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1477
MC68HC11A8	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1518
MC68HC11D3	8-Bit HCMOS Microcontroller Unit SPI, SCI, and EEPROM	3-1560
MC68HC11E1	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1562
MC68HC11E9	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1563
MC68HC11F1	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1606
MC68HC711D3	8-Bit HCMOS Microcontroller Unit with OTPROM or EEPROM	3-1608
MC68HC811E2	8-Bit HCMOS Microcontroller Unit with SPI, SCI, A/D Converter, and EEPROM	3-1610
MC146818	Real Time Clock with RAM	3-1653
MC146818A	Real Time Clock with RAM	3-1673
MC6821	Peripheral Interface Adapter	3-1692
MC146823	CMOS Parallel Interface	3-1703
MC68HC24	Port Replacement Module	3-1719
MC68HC34	Dual Port RAM	3-1736
MC6840	Programmable Timer Module	3-1744
MC6844	Direct Memory Access Controller	3-1757
MC6845	CRT Controller	3-1774

MASTER INDEX (Concluded)

Device Number	Description	Page Number
MC6850	Asynchronous Communications Interface Adapter	3-1798
MC6852	Synchronous Serial Data Adapter	3-1807
MC6854	Advanced Data Link Controller	3-1821
MC68488	General-Purpose Interface Adapter	3-1844
MC6898	Cable Driver and Receiver	3-1864
MC68HC99	Hard Disk Controller	3-1870

68HC811-BYCJC
CLCC 52 PIN

2K EPROM
192 BYTES RAM

Data Sheets

Volume I and II

3

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC05C9

Product Preview

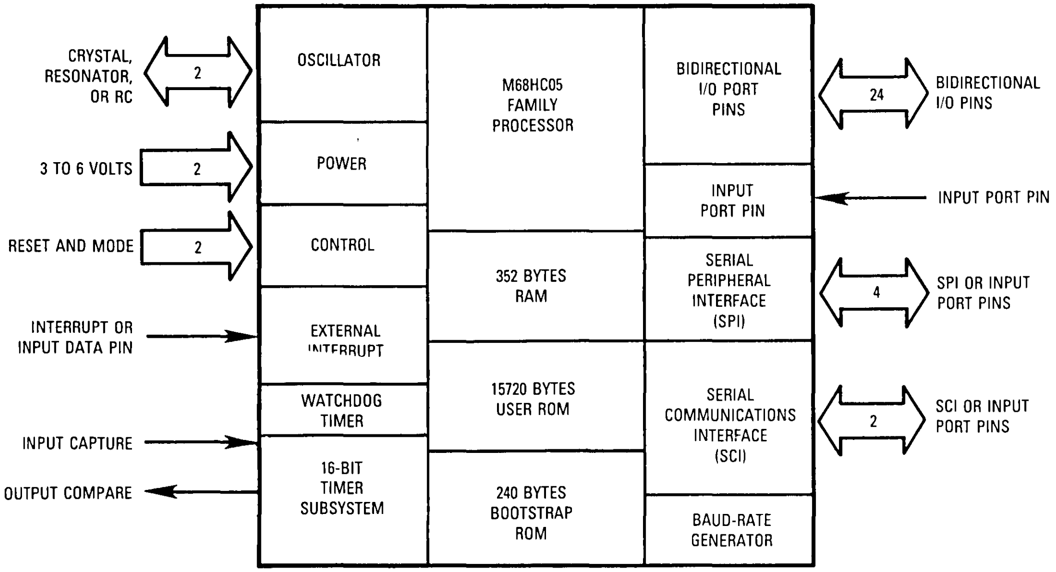
8-Bit Microcontroller Unit

The MC68HC05C9 is an advanced 8-bit microcontroller unit (MCU) with highly sophisticated on-chip peripheral capabilities. It is similar to the MC68HC705C8 with some differences including 15720 bytes of ROM and 352 bytes of RAM.

The following are some of the hardware and software features of the MC68HC05C9.

- HCMOS Technology
- Power Saving STOP, WAIT, and Data Retention Modes
- Fully Static Operation
- 15720 Bytes of User ROM
- 352 Bytes of RAM
- 240 Bytes of Bootstrap ROM
- 24 Bidirectional I/O Lines
- 16-Timer Subsystem
- Serial Communications Interface System (SCI)
- Serial Peripheral Interface System (SPI)
- Interrupts: External, Timer, SCI, and SPI
- Master Reset and Power-On Reset
- 2.1 MHz Internal Operation Frequency at 5 Volts; 1.0 MHz at 3 Volts

BLOCK DIAGRAM


CRYSTAL, RESONATOR, OR RC
2
3 TO 6 VOLTS
2
RESET AND MODE
2
INTERRUPT OR INPUT DATA PIN
INPUT CAPTURE
OUTPUT COMPARE
OSCILLATOR
POWER
CONTROL
EXTERNAL INTERRUPT
WATCHDOG TIMER
16-BIT TIMER SUBSYSTEM
M68HC05 FAMILY PROCESSOR
352 BYTES RAM
15720 BYTES USER ROM
240 BYTES BOOTSTRAP ROM
BIDIRECTIONAL I/O PORT PINS
INPUT PORT PIN
SERIAL PERIPHERAL INTERFACE (SPI)
SERIAL COMMUNICATIONS INTERFACE (SCI)
BAUD-RATE GENERATOR
24
BIDIRECTIONAL I/O PINS
INPUT PORT PIN
4
SPI OR INPUT PORT PINS
2
SCI OR INPUT PORT PINS

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

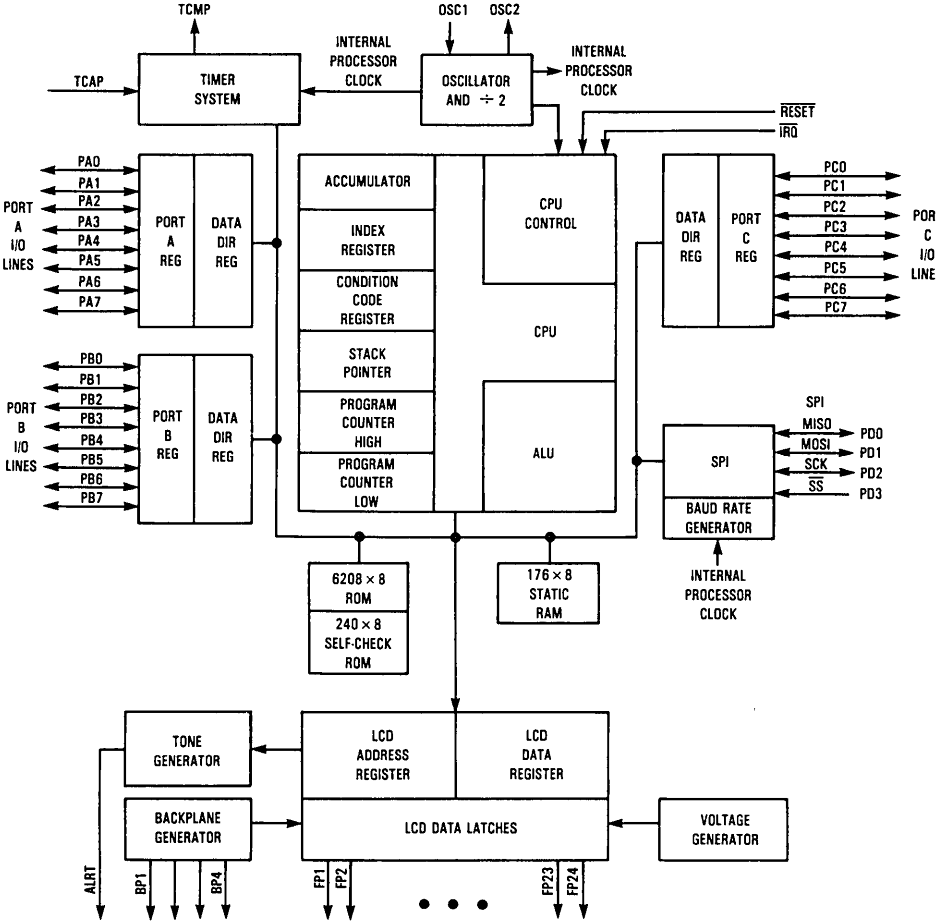
MC68HC05L6

Technical Summary
8-Bit Microcontroller Unit

The MC68HC05L6 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are shown on page 2.

BLOCK DIAGRAM


TCMP
TCAP
TIMER SYSTEM
INTERNAL PROCESSOR CLOCK
OSC1
OSC2
OSCILLATOR AND ÷ 2
INTERNAL PROCESSOR CLOCK
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A I/O LINES
PORT A REG
DATA DIR REG
ACCUMULATOR
INDEX REGISTER
CONDITION CODE REGISTER
STACK POINTER
PROGRAM COUNTER HIGH
PROGRAM COUNTER LOW
CPU CONTROL
CPU
ALU
DATA DIR REG
PORT C REG
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PORT C I/O LINES
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PORT B I/O LINES
PORT B REG
DATA DIR REG
SPI
MISO
MOSI
SCK
SS
PD0
PD1
PD2
PD3
SPI
BAUD RATE GENERATOR
INTERNAL PROCESSOR CLOCK
6208 × 8 ROM
240 × 8 SELF-CHECK ROM
176 × 8 STATIC RAM
TONE GENERATOR
LCD ADDRESS REGISTER
LCD DATA REGISTER
BACKPLANE GENERATOR
LCD DATA LATCHES
VOLTAGE GENERATOR
ALRT
BP1
BP4
FP1
FP2
FP23
FP24

This document contains information on a new product. Specifications and information herein are subject to change without notice.

Hardware Features

- HCMOS Technology
- 8-Bit Architecture
- Power-Saving Stop and Wait Modes
- RAM and CPU Register Contents Valid at $V_{CC}=2.0$ Volts (CPU Halted)
- Independent Power Supplies (CPU – LCD) 3- to 6-Volt Operation
- 176 Bytes of On-Chip RAM
- 6208 Bytes of On-Chip ROM
- 24 Bidirectional I/O Lines
- 4.0 MHz Internal Operating Frequency at 5 Volts
- Internal 16-Bit Timer Similar to MC6801 Timer
- Serial Peripheral Interface System
- Self-Check Mode
- External, Timer, and Serial Peripheral Interface Interrupts
- Master Reset and Power-On Reset
- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- LCD 2/3, 1/3 V_{LL} Divider Circuitry Included
- Tone Generator
- 68-Pin Quad Package

Software Features

- Similar to MC6800
- 8×8 Unsigned Multiply Instruction
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Programmable LCD Driver Circuitry
  - 96 Pixels in ×4 Mode
  - 72 Pixels in ×3 Mode
- Low-Cost Development Support with Evaluation Module

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD}, V_{LL} AND V_{SS}

Power is supplied to the microcontroller using these pins. V_{DD} is the positive supply for the processor, V_{LL} is power for the LCD driver circuits, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

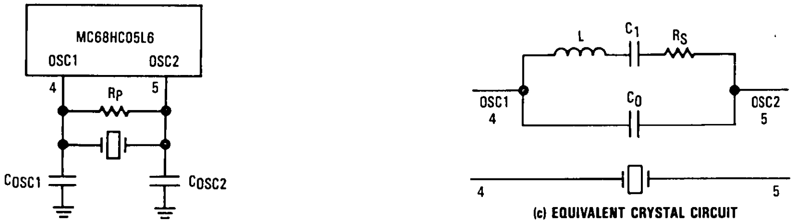
CRYSTAL

	2 MHz	4 MHz	UNITS
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15–40	15–30	pF
C_{OSC2}	15–30	15–25	pF
R_P	10	10	MΩ
Q	30	40	K

CERAMIC RESONATOR

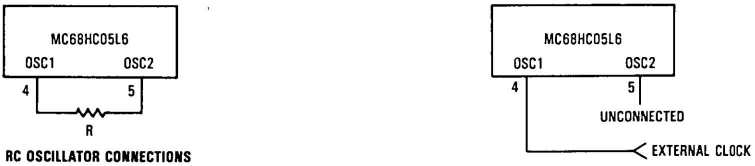
	2–4 MHz	UNITS
R_S (TYPICAL)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1–10	MΩ
Q	1250	–

(a) CRYSTAL/CERAMIC RESONATOR PARAMETERS


MC68HC05L6
OSC1
OSC2
4
5
RP
COSC1
COSC2
L
C1
RS
C0
OSC1
4
OSC2
5
4
5

(b) CRYSTAL/CERAMIC RESONATOR OSCILLATOR CONNECTIONS

(c) EQUIVALENT CRYSTAL CIRCUIT


MC68HC05L6
OSC1
OSC2
4
5
R
MC68HC05L6
OSC1
OSC2
4
5
UNCONNECTED
EXTERNAL CLOCK

(d) RC OSCILLATOR CONNECTIONS

(e) EXTERNAL CLOCK SOURCE CONNECTIONS (EITHER CRYSTAL OR RC MASK OPTIONS)

Figure 1. Oscillator Connections

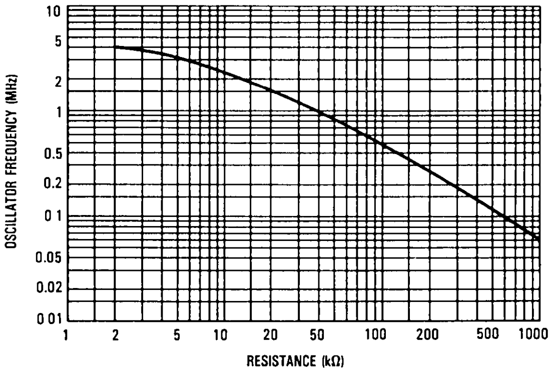


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD3)

These four lines comprise port D, a fixed input port. Any special functions that are enabled (SPI) affect this port. Refer to **PROGRAMMING** for additional information.

BP1-BP4

These four output lines provide the backplane drive signals to the liquid crystal display unit.

FP1-FP24

These 24 output lines provide the frontplane drive signals to the liquid crystal display unit.

ALRT

This pin provides the tone generator output signal.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

$R/\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read
1	1	The I/O pin is in an output mode. The output data latch is read.

*$R/\overline{W}$ is an internal signal

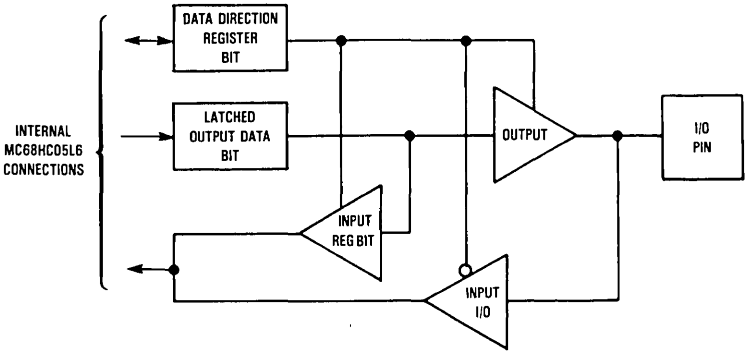


Figure 3. Typical Port I/O Circuit

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD3) that monitors the external pins whenever the SPI is disabled. After reset, all four bits become valid inputs because the special function drivers are disabled. For example, with the SPI enabled, PD0–PD3 inputs will read zero. With the SPI disabled, PD0–PD3 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SPI) PROGRAMMING

The SPI uses the port D pins for its functions. The SPI function requires four of the pins (PD0-PD3) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, LCD drivers, and I/O. The user-defined reset and interrupt vectors are located from $1FF0 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

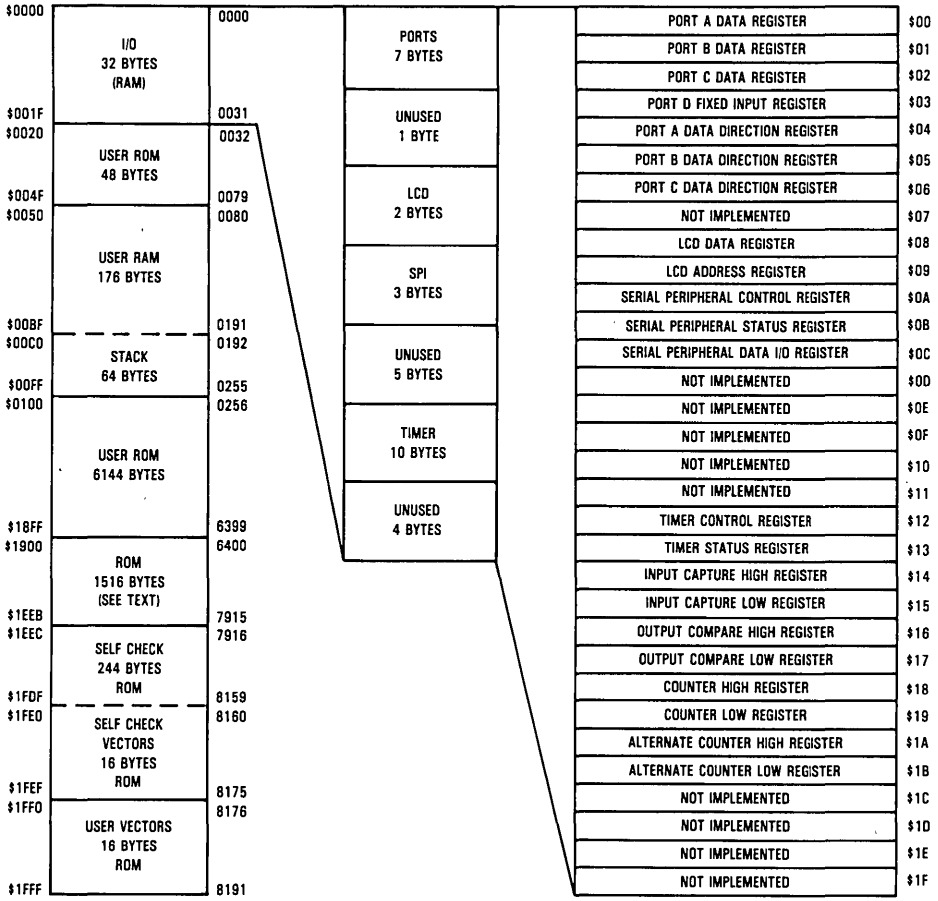

$0000
0000
I/O
32 BYTES
(RAM)
$001F
0031
$0020
0032
USER ROM
48 BYTES
$004F
0079
$0050
0080
USER RAM
176 BYTES
$00BF
0191
$00C0
0192
STACK
64 BYTES
$00FF
0255
$0100
0256
USER ROM
6144 BYTES
$18FF
6399
$1900
6400
ROM
1516 BYTES
(SEE TEXT)
$1EEB
7915
$1EEC
7916
SELF CHECK
244 BYTES
ROM
$1FDF
8159
$1FE0
8160
SELF CHECK
VECTORS
16 BYTES
ROM
$1FEF
8175
$1FF0
8176
USER VECTORS
16 BYTES
ROM
$1FFF
8191
PORTS
7 BYTES
UNUSED
1 BYTE
LCD
2 BYTES
SPI
3 BYTES
UNUSED
5 BYTES
TIMER
10 BYTES
UNUSED
4 BYTES
PORT A DATA REGISTER $00
PORT B DATA REGISTER $01
PORT C DATA REGISTER $02
PORT D FIXED INPUT REGISTER $03
PORT A DATA DIRECTION REGISTER $04
PORT B DATA DIRECTION REGISTER $05
PORT C DATA DIRECTION REGISTER $06
NOT IMPLEMENTED $07
LCD DATA REGISTER $08
LCD ADDRESS REGISTER $09
SERIAL PERIPHERAL CONTROL REGISTER $0A
SERIAL PERIPHERAL STATUS REGISTER $0B
SERIAL PERIPHERAL DATA I/O REGISTER $0C
NOT IMPLEMENTED $0D
NOT IMPLEMENTED $0E
NOT IMPLEMENTED $0F
NOT IMPLEMENTED $10
NOT IMPLEMENTED $11
TIMER CONTROL REGISTER $12
TIMER STATUS REGISTER $13
INPUT CAPTURE HIGH REGISTER $14
INPUT CAPTURE LOW REGISTER $15
OUTPUT COMPARE HIGH REGISTER $16
OUTPUT COMPARE LOW REGISTER $17
COUNTER HIGH REGISTER $18
COUNTER LOW REGISTER $19
ALTERNATE COUNTER HIGH REGISTER $1A
ALTERNATE COUNTER LOW REGISTER $1B
NOT IMPLEMENTED $1C
NOT IMPLEMENTED $1D
NOT IMPLEMENTED $1E
NOT IMPLEMENTED $1F

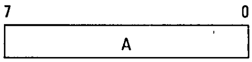
Figure 4. Memory Map

REGISTERS

The MCU contains the registers described in the following paragraphs.

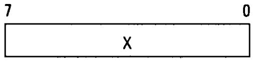
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



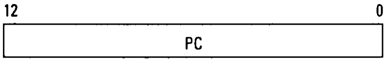
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

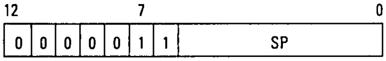
The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

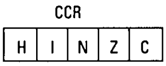
The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF flag
Interrupts — Tests external, timer and SPI interrupts
LCD — Displays the SC number
SPI — Transmission test; checks SPIF, WCOL, and MODF flags

Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

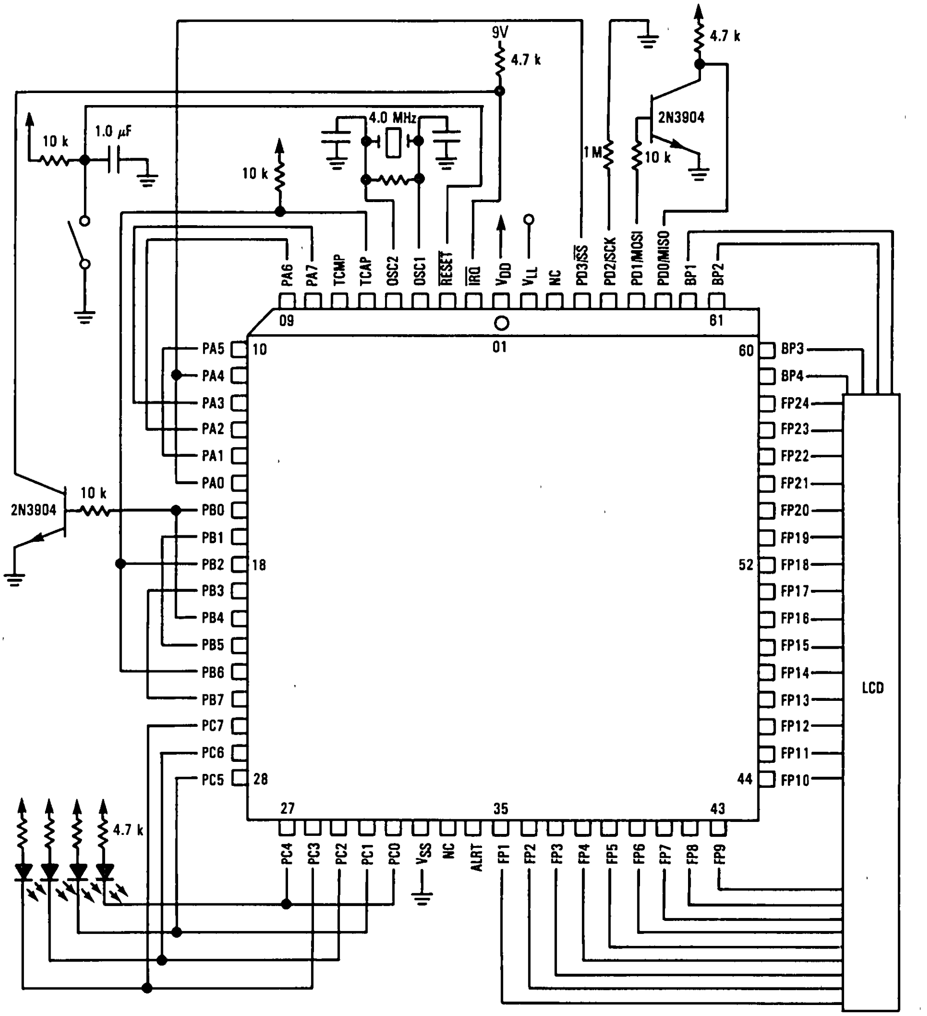

9V
4.7 k
4.7 k
10 k
1.0 μF
4.0 MHz
10 k
1 M
10 k
2N3904
2N3904
10 k
4.7 k
PA6
PA7
TCMP
TCAP
OSC2
OSC1
RESET
IRQ
VDD
VLL
NC
PD3/SS
PD2/SCK
PD1/MOSI
PD0/MISO
BP1
BP2
09
01
61
PA5
PA4
PA3
PA2
PA1
PA0
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC7
PC6
PC5
10
18
28
60
52
44
BP3
BP4
FP24
FP23
FP22
FP21
FP20
FP19
FP18
FP17
FP16
FP15
FP14
FP13
FP12
FP11
FP10
LCD
27
35
43
PC4
PC3
PC2
PC1
PC0
VSS
NC
ALRT
FP1
FP2
FP3
FP4
FP5
FP6
FP7
FP8
FP9

Figure 5. Self-Check Circuit Schematic Diagram

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad ROM
1	1	0	1	Bad SPI
1	1	1	0	Bad Interrupts or IRQ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on; 1 indicates LED is off.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM checksum subroutine is called at location \$1F7D with RAM location \$0053 equal to \$01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations \$0050 through \$0053 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the RESET line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{CYC}) delay after the oscillator becomes active. If the RESET pin is low at the end of 4046 t_{CYC}, the MCU will remain in the reset condition until RESET goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period of one and one-half machine cycles (t_{CYC}).

INTERRUPTS

The MCU can be interrupted four different ways: the three maskable hardware interrupts (IRQ, SPI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at IRQ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

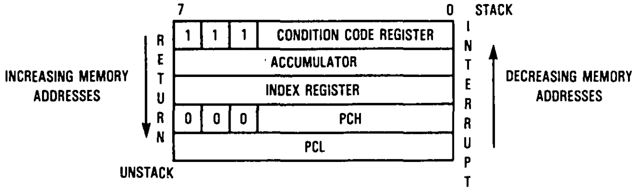
Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 6. Interrupt Stacking Order

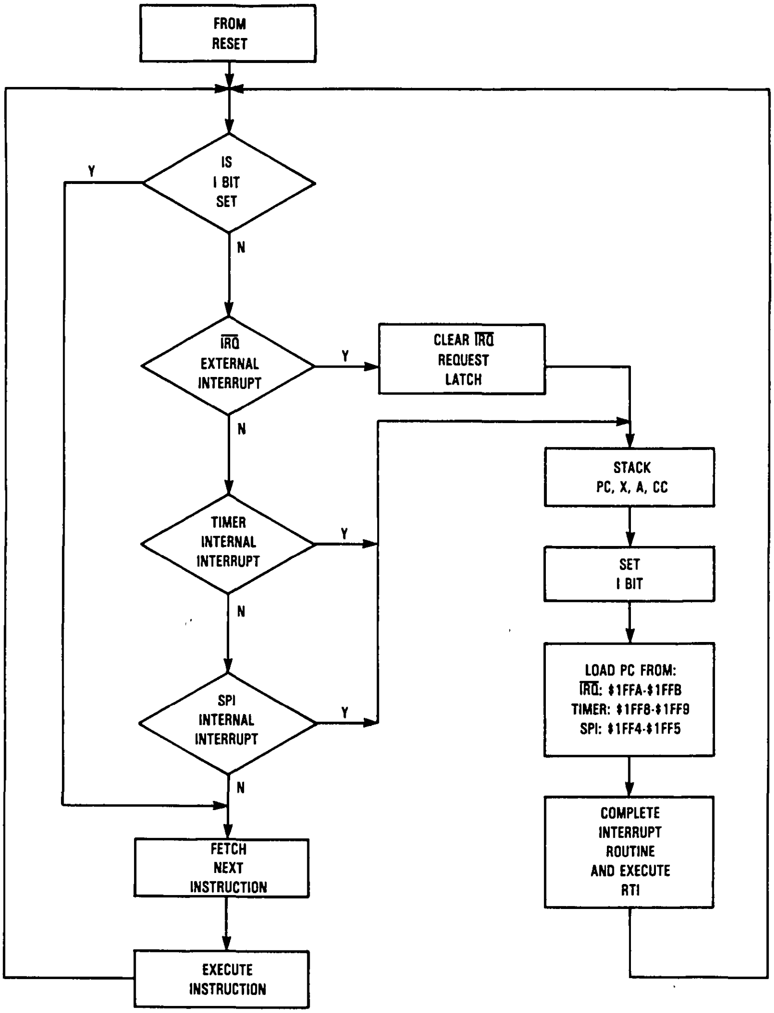
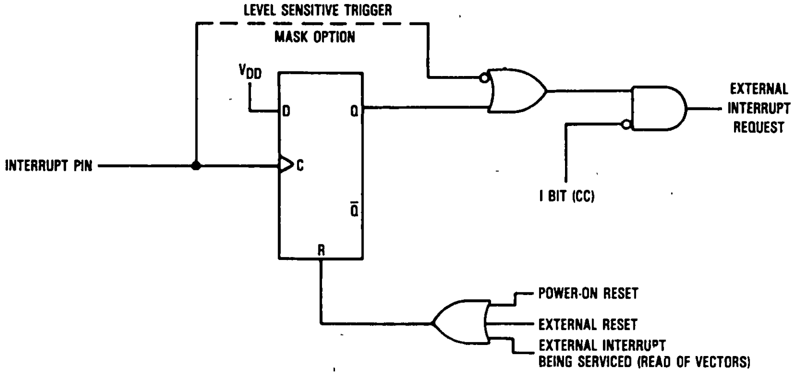
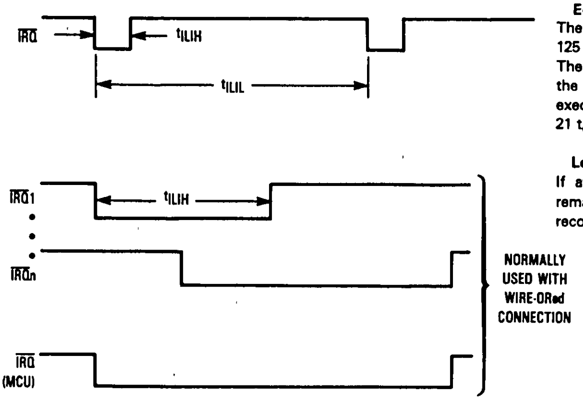

FROM RESET
IS I BIT SET
Y
N
$\overline{\text{IRQ}}$ EXTERNAL INTERRUPT
Y
N
CLEAR $\overline{\text{IRQ}}$ REQUEST LATCH
TIMER INTERNAL INTERRUPT
Y
N
SPI INTERNAL INTERRUPT
Y
N
STACK PC, X, A, CC
SET I BIT
LOAD PC FROM:
$\overline{\text{IRQ}}$: $1FFA-$1FFB
TIMER: $1FF8-$1FF9
SPI: $1FF4-$1FF5
COMPLETE INTERRUPT ROUTINE AND EXECUTE RTI
FETCH NEXT INSTRUCTION
EXECUTE INSTRUCTION

Figure 7. Reset and Interrupt Processing Flowchart


LEVEL SENSITIVE TRIGGER
MASK OPTION
V_{DD}
D
Q
C
Q̄
R
INTERRUPT PIN
EXTERNAL INTERRUPT REQUEST
I BIT (CC)
POWER-ON RESET
EXTERNAL RESET
EXTERNAL INTERRUPT BEING SERVICED (READ OF VECTORS)

(a) INTERRUPT FUNCTION DIAGRAM


IRQ
t_{ILIH}
t_{ILIL}
IRQ1
IRQn
IRQ (MCU)
NORMALLY USED WITH WIRE-ORed CONNECTION

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns ($V_{DD}=5$ V) or 250 ns ($V_{DD}=3$ V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles.

Level-Sensitive Trigger Condition
If after servicing an interrupt the IRQ remains low, then the next interrupt is recognized.

(b) INTERRUPT MODE DIAGRAM

Figure 8. External Interrupt

is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, LCD, and SPI operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the IRQ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the $\overline{IRQ}$ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

LCD during Stop Mode

When the MCU enters the STOP mode, the LCD is disabled. The rate generator stops, the internal voltage generator is shut down, and all LCD pins are pulled to V_{SS}. All LCD data is retained. If the IRQ pin is used to exit the STOP mode, the LCD display can be re-enabled with a bit in the LCD address register. If a reset is used to exit the STOP mode, the LCD logic must be reconfigured before it can be used.

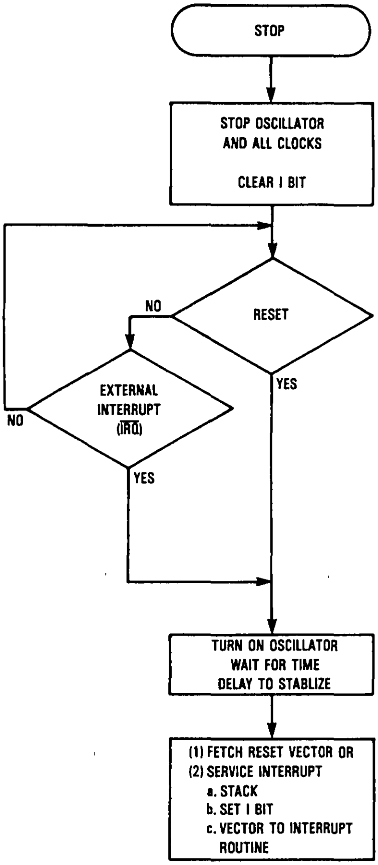

STOP
STOP OSCILLATOR AND ALL CLOCKS
CLEAR I BIT
RESET
NO
YES
EXTERNAL INTERRUPT ($\overline{IRQ}$)
NO
YES
TURN ON OSCILLATOR WAIT FOR TIME DELAY TO STABLIZE
(1) FETCH RESET VECTOR OR
(2) SERVICE INTERRUPT
a. STACK
b. SET I BIT
c. VECTOR TO INTERRUPT ROUTINE

Figure 9. STOP Function Flowchart

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, LCD, and SPI remain active (refer to Figure 10). An interrupt from the timer, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{RESET}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, \$18–\$19 (counter register) or \$1A–\$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter (\$19, \$1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate

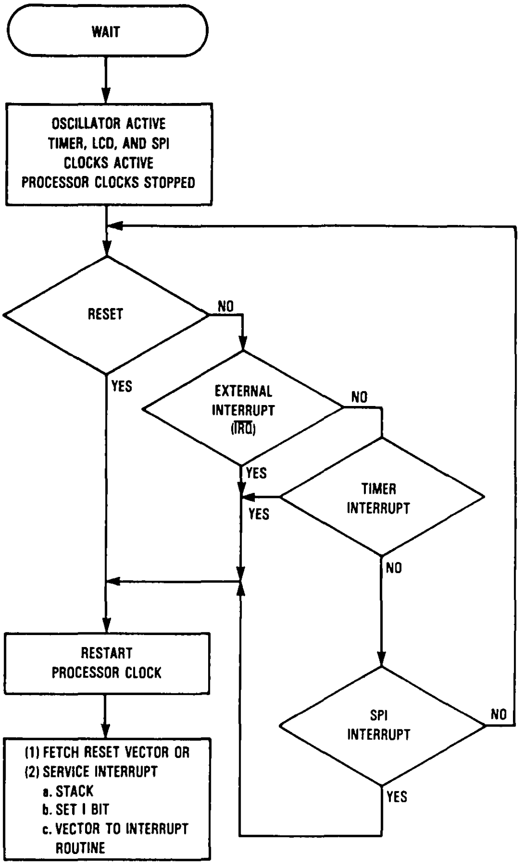

WAIT
OSCILLATOR ACTIVE
TIMER, LCD, AND SPI
CLOCKS ACTIVE
PROCESSOR CLOCKS STOPPED
RESET
NO
YES
EXTERNAL
INTERRUPT
(IRQ)
NO
YES
TIMER
INTERRUPT
YES
NO
RESTART
PROCESSOR CLOCK
(1) FETCH RESET VECTOR OR
(2) SERVICE INTERRUPT
a. STACK
b. SET I BIT
c. VECTOR TO INTERRUPT
ROUTINE
SPI
INTERRUPT
NO
YES

Figure 10. WAIT Function Flowchart

register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually,

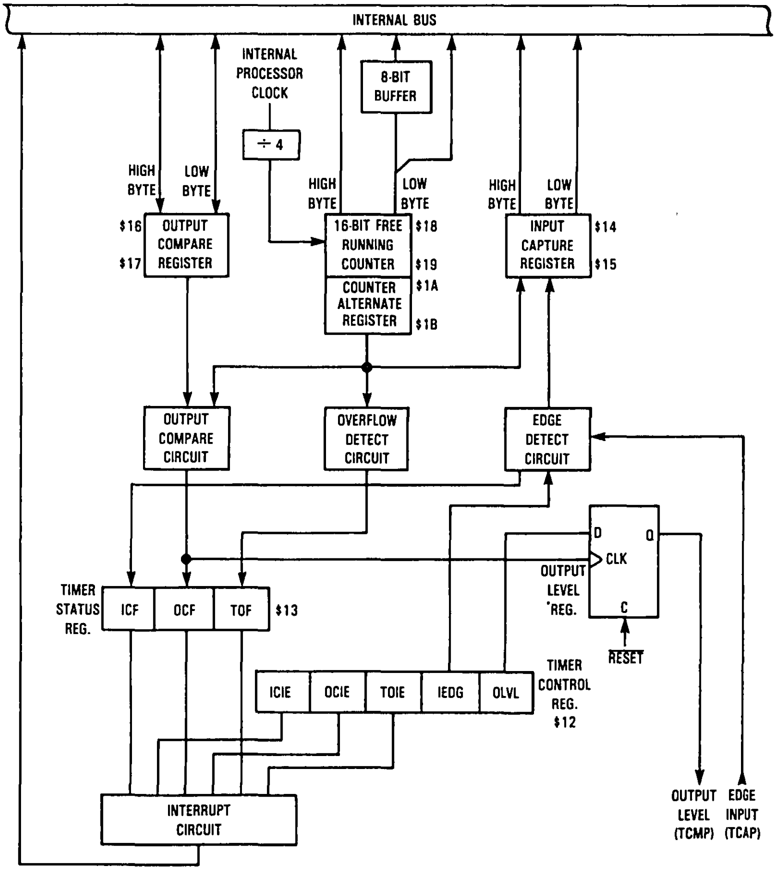

INTERNAL BUS
INTERNAL PROCESSOR CLOCK
8-BIT BUFFER
÷ 4
HIGH BYTE
LOW BYTE
HIGH BYTE
LOW BYTE
HIGH BYTE
LOW BYTE
$16
$17
OUTPUT COMPARE REGISTER
16-BIT FREE RUNNING COUNTER
$18
$19
COUNTER ALTERNATE REGISTER
$1A
$1B
INPUT CAPTURE REGISTER
$14
$15
OUTPUT COMPARE CIRCUIT
OVERFLOW DETECT CIRCUIT
EDGE DETECT CIRCUIT
D
Q
CLK
C
OUTPUT LEVEL REG.
RESET
TIMER STATUS REG.
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
TIMER CONTROL REG.
$12
INTERRUPT CIRCUIT
OUTPUT LEVEL (TCMP)
EDGE INPUT (TCAP)

Figure 11. Timer Block Diagram

and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled
OCIE — Output Compare Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled
TOIE — Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled
IEDG — Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG bit (U = unaffected).
OLVL — Output Level
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
1 = High output
0 = Low output
Bits 2, 3, and 4 — Not used
Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET.	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed
OCF — Output Compare Flag
1 = Flag set when output compare register contents match the free-running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed
TOF — Timer Overflow Flag
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed
Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

LIQUID CRYSTAL DISPLAY DRIVER AND TONE GENERATOR

This MCU contains liquid crystal driver (LCD) circuits and a tone generator.

LCD DRIVER

Figure 12 is a functional block diagram of the LCD driver circuit. The LCD address selects one of the 12 LCD data latches which the processor can read or write. The LCD data register will contain the data that is in the addressed latch. The data can either be read by the processor, or new data can be written.

The first write to the LCD address register sets the configuration register. This data must contain the following information:

Fast charge on or off
Selection of ×3 or ×4 multiplexing
Selection of voltage generator
Selection of crystal/LCD operating frequency ratio

After the first write, the LCD address register retains only address information, and the configuration register is not modified.

The LCD address register data is a binary number ranging from zero to eleven. A pointer associated with the address register selects the LCD data latch for each digit. As soon as a latch is selected, the data in that latch is available at the LCD data register ($08) for use by the processor. The LCD data latches are dual port, accessible from both the LCD data register and the scan logic. While the processor is reading or writing the LCD data register, the data in the addressed latch is also being read by the scan logic circuits. The circuits read the contents of the LCD data latches and move that data to the LCD driver circuit where the proper frontplane (FPx) and backplane (BPx) outputs are generated to select the desired pixels.

Figure 13 is a map of the relationship of the individual backplane/frontplane pins to the individual LCD data latch bit positions. This is shown for ×4 multiplexing. in ×3 multiplexing, each BP4 position is not available.

Voltage Generator

The voltage generator divides the LCD voltage (V_{LL}) into the 2/3, 1/3 voltage components required by multiplexed LCDs. These voltage levels are supplied to the driver circuits. This MCU has two voltage generators (high and low power) for use with large or small displays. Additional circuits supply even higher current for the fast-charge option.

The fast-charge option is used to quickly charge the capacitance of very large displays by using an extra high-current voltage generator for a very short period after each backplane or frontplane signal edge. The voltage generator also contains additional supplies that are used for larger displays. The fast-charge option is available for both the high- and low-power voltage generators.

Tone Generator

The tone generator is a counter that uses the crystal frequency input to produce an audio frequency square wave at the ALRT pin. (The ratio between the processor clock frequency and the tone frequency is determined by a user mask option.) The controls for the tone generator are located in the LCD address register. Figure 14 is a simplified diagram of the tone generator. The crystal frequency table shown in Figure 14 provides a list of the tone output frequencies provided by some of the more common crystal frequencies and the four divider options available.

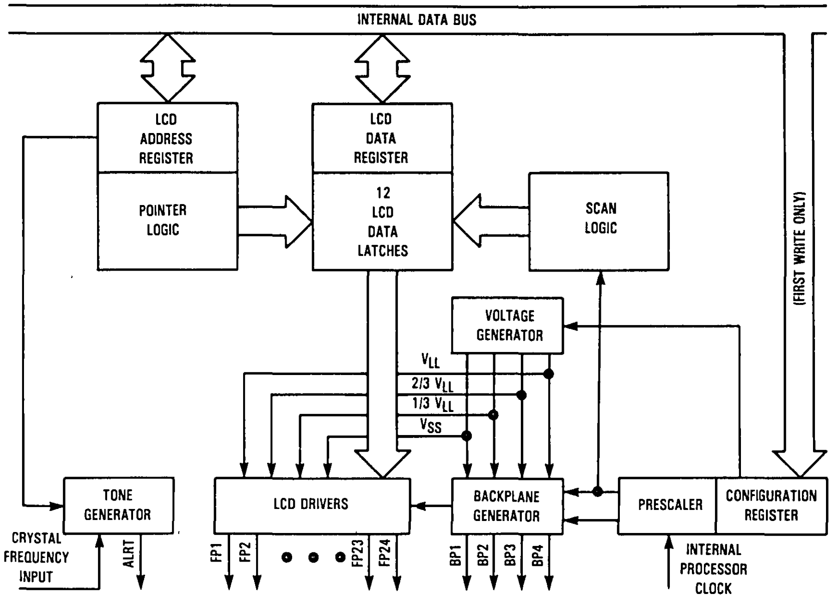


Figure 12. LCD Driver Circuits — Functional Block Diagram

LCD DATA LATCH 00	7	6	5	4	3	2	1	0
($00)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP02				FP01			

LCD DATA LATCH 01	7	6	5	4	3	2	1	0
($01)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP04				FP03			

LCD DATA LATCH 02	7	6	5	4	3	2	1	0
($02)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP06				FP05			

LCD DATA LATCH 03	7	6	5	4	3	2	1	0
($03)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP08				FP07			

LCD DATA LATCH 04	7	6	5	4	3	2	1	0
($04)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP10				FP09			

LCD DATA LATCH 05	7	6	5	4	3	2	1	0
($05)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP12				FP11			

LCD DATA LATCH 06	7	6	5	4	3	2	1	0
($06)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP14				FP13			

LCD DATA LATCH 07	7	6	5	4	3	2	1	0
($07)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP16				FP15			

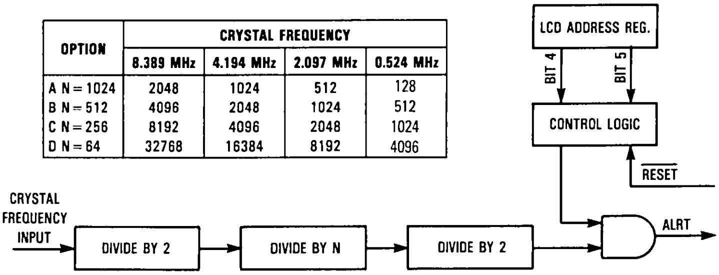
LCD DATA LATCH 08	7	6	5	4	3	2	1	0
($08)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP18				FP17			

LCD DATA LATCH 09	7	6	5	4	3	2	1	0
($09)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP20				FP19			

LCD DATA LATCH 10	7	6	5	4	3	2	1	0
($0A)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP22				FP21			

LCD DATA LATCH 11	7	6	5	4	3	2	1	0
($0B)	BP4	BP3	BP2	BP1	BP4	BP3	BP2	BP1
	FP24				FP23			

Figure 13. Relationship of Backplane/Frontplane Pin to LCD Data Latch Bit



OPTION	CRYSTAL FREQUENCY			
	8.389 MHz	4.194 MHz	2.097 MHz	0.524 MHz
A N=1024	2048	1024	512	128
B N=512	4096	2048	1024	512
C N=256	8192	4096	2048	1024
D N=64	32768	16384	8192	4096

Figure 14. Tone Generator Simplified Diagram

LCD Address Register $0009

The LCD address register points to one of the twelve data latches.

7	6	5	4	3	2	1	0
LON	LOF	TON	TOF	AR3	AR2	AR1	AR0
RESET							
0	0	0	0	0	0	0	0

LON — LCD Drivers On

Writing a one to this bit turns on the LCD drivers. This bit always reads zero.

LOF — LCD Drivers Off

Writing a one to this bit turns off the LCD drivers. Reset also disables the LCD drivers. This bit always reads zero.

TON — Tone Generator On

Writing a one to this bit turns on the tone generator. This bit always reads zero.

TOF — Tone Generator Off

Writing a one to this bit turns off the tone generator. Reset also turns off the tone generator. This bit always reads zero.

AR3–AR0 — Address Register Select

A binary number from zero to 11 written in these bits selects the data latch to be used.

LCD Configuration Register $0009

The LCD configuration register is available only for the first write after a reset at the same address as the LCD address register.

7	6	5	4	3	2	1	0
EFC	×3	VGN	FR4	FR3	FR2	FR1	FR0
RESET							
0	0	0	0	0	0	0	0

EFC — Enable Fast Charge
1 = Fast charge enabled
0 = Fast charge disabled

×3 — Times 3 Multiplexing Mode
1 = Times 3 multiplexing enabled
0 = Times 4 multiplexing enabled

VGN — Voltage Generator Select
1 = High-current voltage generator selected
0 = Low-current voltage generator selected

FR4–FR0 — Operating Frequency Ratio

A binary number in these bits selects the operating frequency ratio as shown in Table 3.

LCD Data Register $0008

The LCD data register accepts data from the processor that goes to the LCD data latch selected by the LCD address register. The LCD data register also contains the data present in the LCD data latch addressed by the LCD address register for reading by the processor. This register is initialized whenever a write to the LCD address register occurs.

LCD Driver and Tone Generator During WAIT Mode

The LCD driver circuits and tone generator are not affected by the WAIT instruction.

LCD Driver and Tone Generator During STOP Mode

The LCD driver circuits and tone generator are disabled by the STOP instruction.

Determining Proper Levels for V_{LL}

The V_{LL} supply, which is the operating voltage for LCD drive circuits, can range from 3 to 6 V, as required by the display in use. The MCU has internal level translators so that the LCD driver voltage (V_{LL}) is independent of the MCU operating voltage (V_{DD}). Either voltage may be higher than the other.

Figure 15 shows a circuit that can be used to determine the optimum V_{LL} level. The final design could use two fixed resistors in place of the potentiometer.

Most LCD displays have a temperature coefficient of −8 mV/°C. A temperature compensated V_{LL} supply is shown in Figure 16.

To determine which voltage generator to use and whether the fast-charge feature is needed, connect the

Table 3. Operating Frequency Ratio

FR4 FR0	Xtal/LCD	Bus/LCD	FR4 FR0	Xtal/LCD	Bus/LCD
00000	512	256	10000	2048	1024
00001	1024	512	10001	4096	2048
00010	2048	1024	10010	8192	4096
00011	4096	2048	10011	16348	8192
00100	8192	4096	10100	32768	16348
00101	16384	8192	10101	65536	32768*
00110	32768	16384	10110	131072	65536
00111	65536	32768*	10111	262144	131072
01000	1536	768	11000	6656	3328
01001	3072	1536	11001	13312	6656
01010	6144	3072	11010	26624	13312
01011	12288	6144	11011	53248	26624
01100	24576	12288	11100	106496	53248
01101	49152	24576	11101	212992	106496
01110	98304	49152	11110	425984	212992
01111	196608	98304	11111	851968	425984

Example: *60-Hz LCD with a 4-MHz crystal.

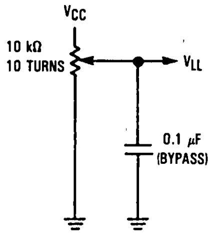


Figure 15. Test Circuit for Determining V_{LL} Drive Level

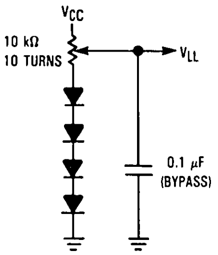


Figure 16. Test Circuit for Incorporating Temperature Compensation in the V_{LL} Supply

MCU and LCD as it will be in the final product (using the final printed circuit board, crystal, etc., if possible) and examine the LCD driver waveform with an oscilloscope and low-capacitance probe. Choose the combination that yields the best waveforms (least noise).

Most displays can be driven with the fast charge disabled. Fast charge is usually required for only very large displays.

LCD Software Examples

The following software segments are examples of loading the registers to perform a total display change, to scroll left, and to scroll right.

NOTE

The parenthetical number in the comment column of the listing represents the number of machine cycles that it takes to execute the given line of code. This information is provided to assist you in determining the time required to execute the examples.

Seven-Segment Display Connections

Figures 17 and 18 provide correlation between the frontplane and backplane outputs of the MCU and a typical seven-segment LCD. Both the ×3 and ×4 multiplexing modes are covered.

Total Display Change

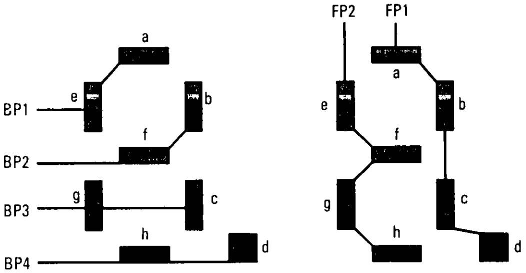
```
           LDX     #11       (2) Initialize the pointer
  LOOP     STX     ADDREG    (4) Point to the desired data register
           LDA     DATA,X    (4) Fetch the data
           STA     DATREG    (4) Store the data
           DEX               (3) Update the data pointer
           BPL     LOOP      (3) Branch unless finished
 11 Bytes, 218 Cycles
```

Scroll Left

```
                LDX    #1       (2) Initialize the pointer
        LOOP1   STX    ADDREG   (4) Point to the desired data register
                LDA    DATREG   (3) Fetch the data
                DEC    ADDREG   (5) Point to the previous data register
                STA    DATREG   (3) Store the data
                INX             (5) Update the data pointer
                CMPX   #12      (2) Are we finished?
                BNE    LOOP1    (3) Branch if no
15 Bytes, 302 Cycles
```

Scroll Right

```
                LDX    #10      (2) Initialize the pointer
        LOOP2   STX    ADDREG   (4) Point to the desired data register
                LDA    DATREG   (3) Fetch the data
                INC    ADDREG   (5) Point to the next data register
                STA    DATREG   (4) Store the data
                DEX             (3) Update the data pointer
                BPL    LOOP2    (3) Branch unless finished
13 Bytes, 266 Cycles
```



SEGMENT TRUTH TABLE*

	FP1	FP2
BP1	a1	e1
BP2	b1	f1
BP3	c1	g1
BP4	d1	h1

*Since there is no standard for backplane and frontplane connections on multiplexed displays, this truth table may be used for this example only.

	FP1	FP2	FP3	FP4		FP23	FP24
BP1	a1	e1	a2	e2		a12	e12
BP2	b1	f1	b2	f2		b12	f12
BP3	c1	g1	c2	g2	. .	c12	g12
BP4	d1	h1	d2	h2		d12	h12
	DIGIT 1		DIGIT 2		DIGIT 12		

Figure 17. Frontplane and Backplane Connections to a Multiplexed-by-Four Seven-Segment LCD (Includes Decimal Point)

Multiplexed Segmented Display Waveforms

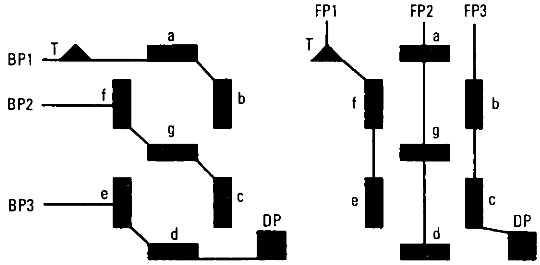
Figure 19 shows examples of the waveforms generated by the LCD driver circuits for application to multiplexed displays.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 20) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase



SEGMENT TRUTH TABLE*

	FP1	FP2	FP3
BP1	T1	a1	b1
BP2	f1	g1	c1
BP3	e1	d1	DP1

*Since there is no standard for backplane and frontplane connections on multiplexed displays, this truth table may be used for this example only

	FP1	FP2	FP3	FP4 FP23	FP24
BP1	T1	a1	b1	T2 . . . . a8	e8
BP2	f1	g1	c1	f2 . . b8	f8
BP3	e1	d1	DP1	e2 c8	g8
	DIGIT 1			DIGIT 2 . . .	DIGIT 8

Figure 18. Frontplane and Backplane Connections to a Multiplexed-by-Three Seven-Segment LCD (Includes Decimal Point and Annunciator)

- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}$ = 1).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 21, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{SS}$) input line selects a slave device. The $\overline{SS}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high; if the $\overline{SS}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{SS}$ line could be tied to V_{SS} as long as CPHA = 1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 22. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin

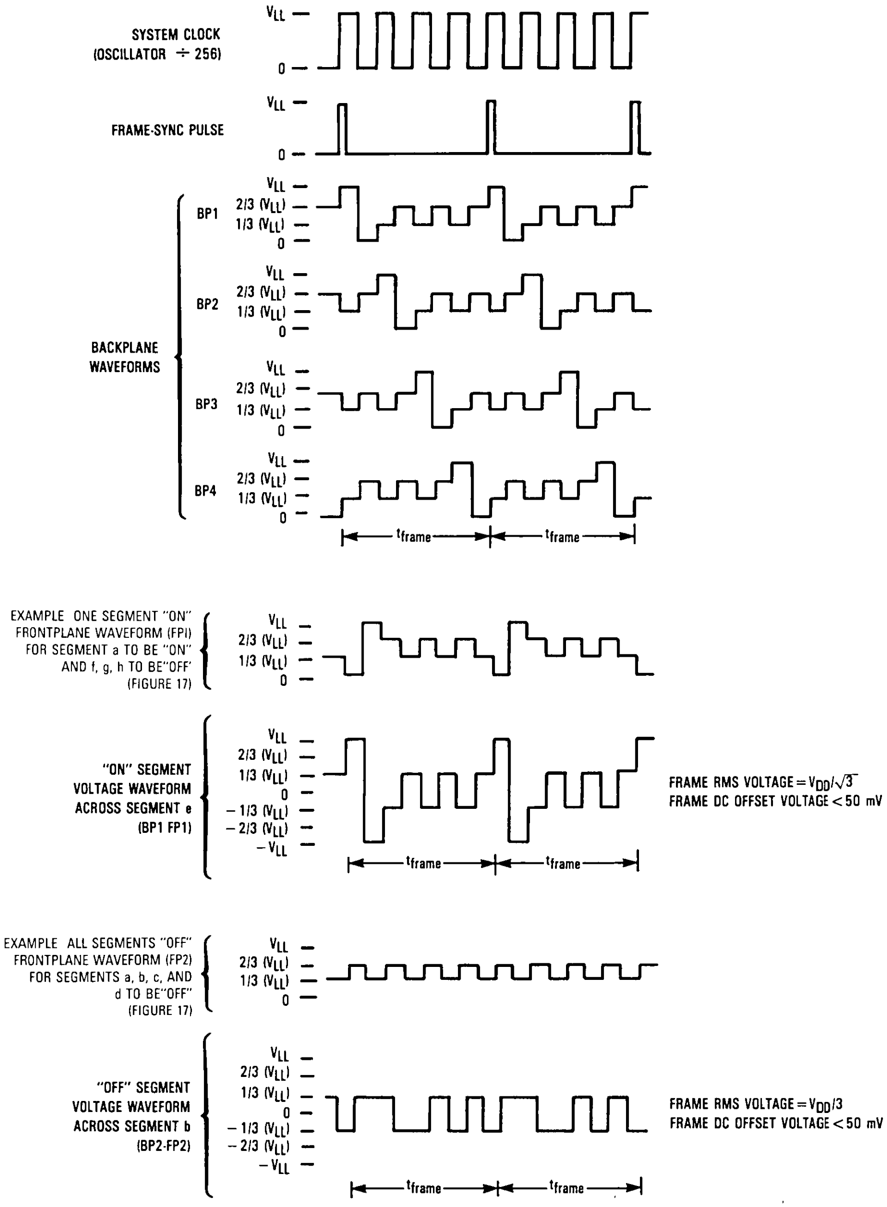

SYSTEM CLOCK
(OSCILLATOR ÷ 256)
V_{LL}
0
FRAME-SYNC PULSE
V_{LL}
0
BACKPLANE WAVEFORMS
BP1
BP2
BP3
BP4
V_{LL}
2/3 (V_{LL})
1/3 (V_{LL})
0
t_{frame}
t_{frame}
EXAMPLE ONE SEGMENT "ON" FRONTPLANE WAVEFORM (FP1) FOR SEGMENT a TO BE "ON" AND f, g, h TO BE "OFF" (FIGURE 17)
"ON" SEGMENT VOLTAGE WAVEFORM ACROSS SEGMENT e (BP1 FP1)
V_{LL}
2/3 (V_{LL})
1/3 (V_{LL})
0
− 1/3 (V_{LL})
− 2/3 (V_{LL})
$-V_{LL}$
FRAME RMS VOLTAGE = $V_{DD}/\sqrt{3}$
FRAME DC OFFSET VOLTAGE < 50 mV
EXAMPLE ALL SEGMENTS "OFF" FRONTPLANE WAVEFORM (FP2) FOR SEGMENTS a, b, c, AND d TO BE "OFF" (FIGURE 17)
"OFF" SEGMENT VOLTAGE WAVEFORM ACROSS SEGMENT b (BP2-FP2)
FRAME RMS VOLTAGE = $V_{DD}/3$
FRAME DC OFFSET VOLTAGE < 50 mV

Figure 19. Multiplexed Segmented Display Waveforms (Sheet 1 of 2)

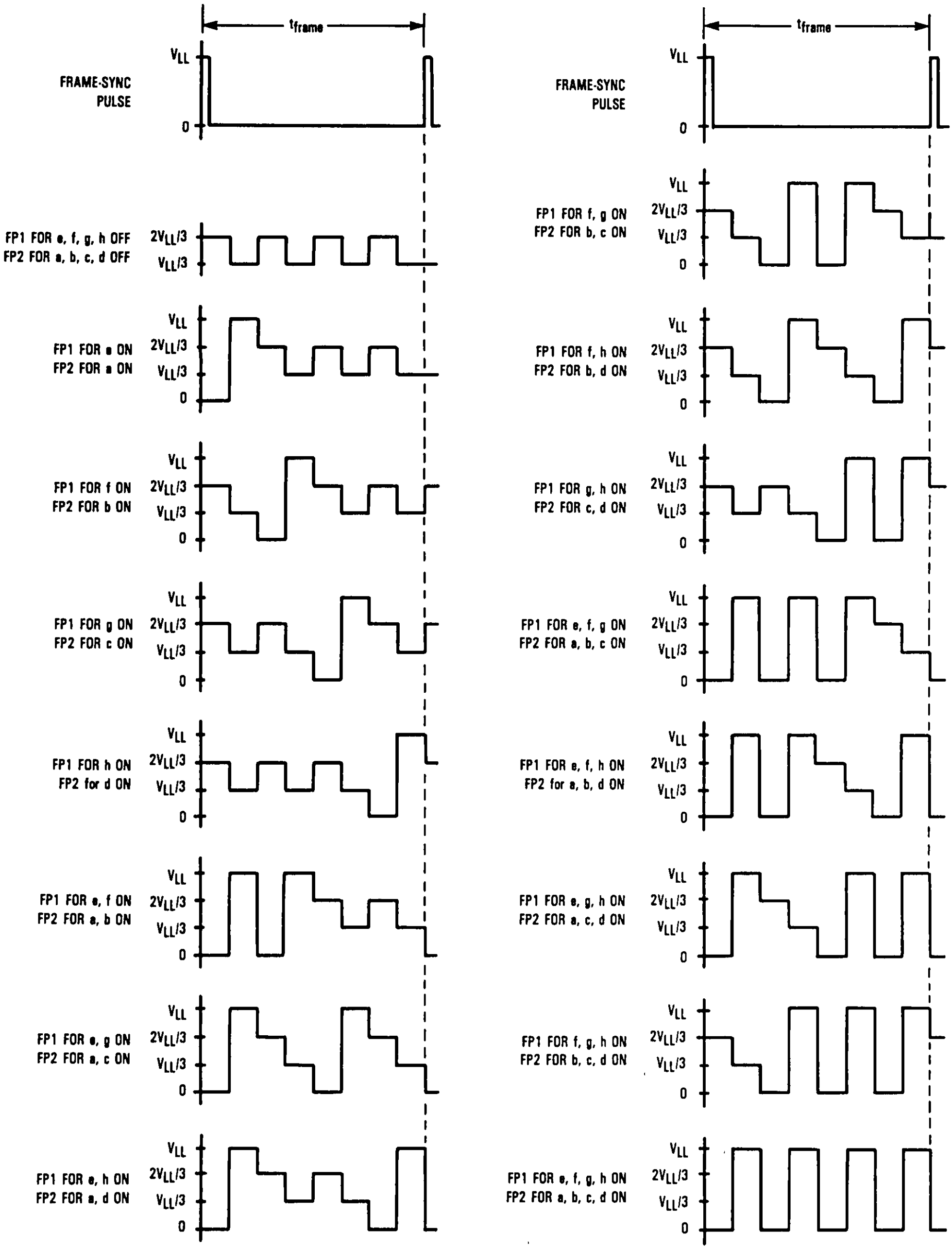
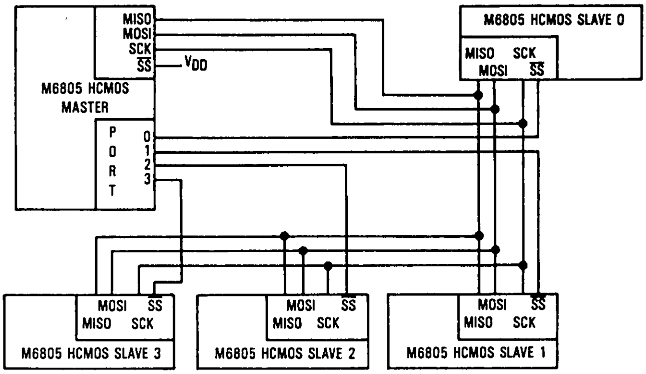

tframe
VLL
FRAME-SYNC PULSE
0
FP1 FOR e, f, g, h OFF
FP2 FOR a, b, c, d OFF
2VLL/3
VLL/3
FP1 FOR e ON
FP2 FOR a ON
FP1 FOR f ON
FP2 FOR b ON
FP1 FOR g ON
FP2 FOR c ON
FP1 FOR h ON
FP2 for d ON
FP1 FOR e, f ON
FP2 FOR a, b ON
FP1 FOR e, g ON
FP2 FOR a, c ON
FP1 FOR e, h ON
FP2 FOR a, d ON
tframe
FRAME-SYNC PULSE
FP1 FOR f, g ON
FP2 FOR b, c ON
FP1 FOR f, h ON
FP2 FOR b, d ON
FP1 FOR g, h ON
FP2 FOR c, d ON
FP1 FOR e, f, g ON
FP2 FOR a, b, c ON
FP1 FOR e, f, h ON
FP2 for a, b, d ON
FP1 FOR e, g, h ON
FP2 FOR a, c, d ON
FP1 FOR f, g, h ON
FP2 FOR b, c, d ON
FP1 FOR e, f, g, h ON
FP2 FOR a, b, c, d ON

Figure 19. Multiplexed Segmented Display Waveforms (Sheet 2 of 2)


MISO
MOSI
SCK
SS
VDD
M6805 HCMOS MASTER
P O R T
0
1
2
3
M6805 HCMOS SLAVE 0
MISO SCK
MOSI SS
MOSI SS
MISO SCK
M6805 HCMOS SLAVE 3
MOSI SS
MISO SCK
M6805 HCMOS SLAVE 2
MOSI SS
MISO SCK
M6805 HCMOS SLAVE 1

(a) SINGLE MASTER, FOUR SLAVES

Figure 20. Master-Slave System Configuration

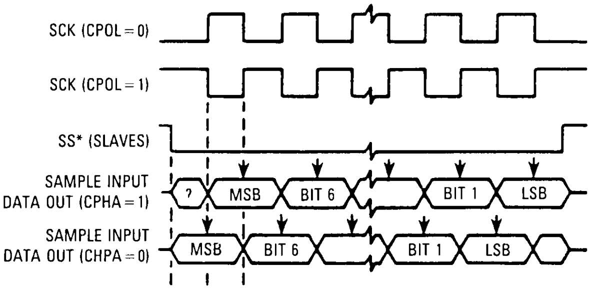

SCK (CPOL=0)
SCK (CPOL=1)
SS* (SLAVES)
SAMPLE INPUT
DATA OUT (CPHA=1)
?
MSB
BIT 6
BIT 1
LSB
SAMPLE INPUT
DATA OUT (CHPA=0)
MSB
BIT 6
BIT 1
LSB

Figure 21. Data Clock Timing Diagram

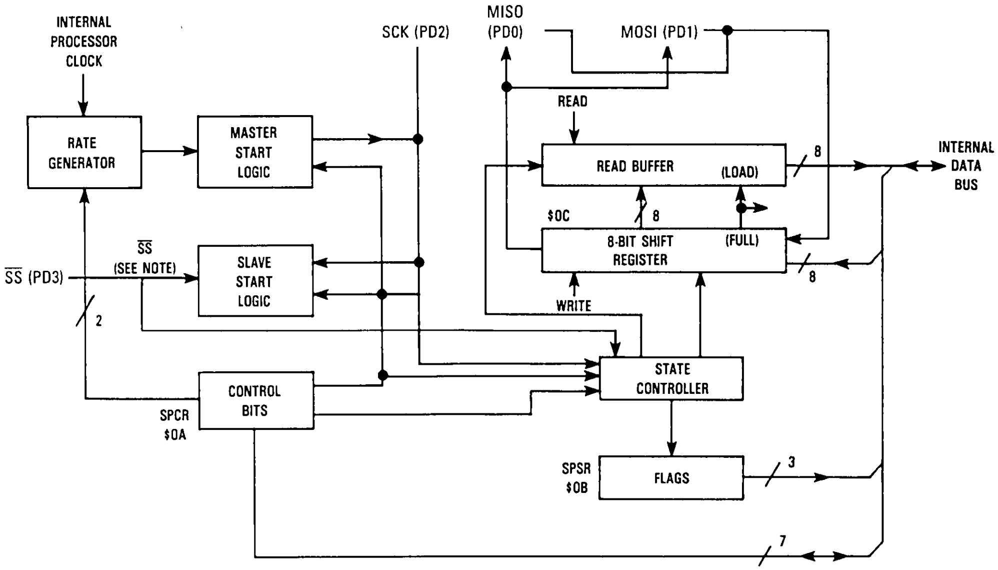

INTERNAL PROCESSOR CLOCK
RATE GENERATOR
MASTER START LOGIC
SCK (PD2)
MISO (PD0)
MOSI (PD1)
READ
READ BUFFER
(LOAD)
8
INTERNAL DATA BUS
$0C
8
8-BIT SHIFT REGISTER
(FULL)
8
SS
(SEE NOTE)
SS (PD3)
SLAVE START LOGIC
WRITE
2
STATE CONTROLLER
CONTROL BITS
SPCR $0A
SPSR $0B
FLAGS
3
7

Figure 22. SPI Block Diagram

and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 23 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

7	6	5	4	3	2	1	0
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0

RESET:

0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
  1 = SPI interrupt enabled
  0 = SPI interrupt disabled
SPE — Serial Peripheral System Enable
  1 = SPI system on
  0 = SPI system off
MSTR — Master Mode Select
  1 = Master mode
  0 = Slave mode
CPOL — Clock Polarity
Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
  1 = SCK line idles high
  0 = SCK line idles in low state
CPHA — Clock Phase
Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
  1 = $\overline{SS}$ is an output enable control.
  0 = Shift clock is the OR of SCK with $\overline{SS}$.
  When $\overline{SS}$ is low, first edge of SCK invokes first data sample.
SPR0, SPR1 — SPI Clock Rate Bits
Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.
Bit 5 - Not used
Can read either one or zero

SPI Clock Rate Selection

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—

RESET

0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
  1 = Indicates data transfer completed between processor and external device.
  (If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
  0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.
WCOL — Write Collision
  1 = Indicates an attempt is made to write to SPDR while data transfer is in process.
  0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.
MODF — Mode Fault Flag
  1 = Indicates multi-master system control conflict.
  0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

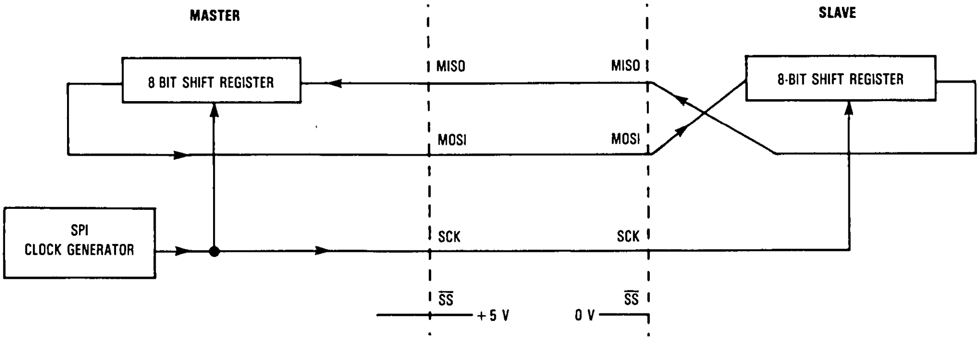


Figure 23. SPI Master-Slave Interconnections

Bits 0–3, and 5 — Not used
Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read/write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register			
Condition Codes	H: Cleared I. Not affected N. Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 4 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

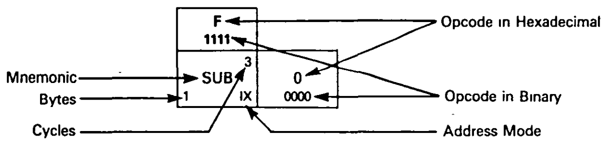
Table 4. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Low / Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
5 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

Legend



RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to 2×V_{DD}+0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05L6	T_A	T_L to T_H 0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Leaded Chip Carrier (PLCC)	θ_{JA}	70	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K - (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 25)	V_{OH}	$V_{DD} - 0.8$	—	—	V
Output Low Voltage (see Figure 26) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 27 and 28)		—	3.5	7.0	mA
Wait (see Figures 27 and 28)		—	1.6	4.0	mA
Stop (see Figure 28)					
25°C		—	2.0	50	µA
0° to 70°C		—	—	140	µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD3	I_{in}	—	—	±1	µA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{RESET}$, $\overline{IRQ}$, TCAP, PD3	C_{in}	—	—	8	

NOTES.

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, active (SPE = TE = RE = 1) add 10% current draw
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is also available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 0.3 V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage	V_{OH}				V
(I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 25)		$V_{DD} - 0.8$ $V_{DD} - 0.3$	— —	— —	
Output Low Voltage (see Figure 26 (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 27 and 29)		—	1.0	2.5	mA
Wait (see Figures 27 and 29)		—	0.5	1.4	mA
Stop (see Figure 29)					
25°C		—	1.0	30	µA
0° to 70°C		—	—	80	µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD3	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD3	I_{in}	—	—	±1	µA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD3, OSC1	C_{in}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0 2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is also available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

V_{DD} = 4.5 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7	3.26 kΩ	2.38 kΩ	50 pF
PD0-PD3	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7	10.91 kΩ	6.32 kΩ	50 pF

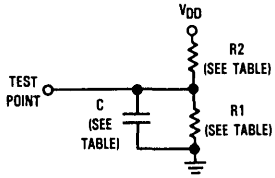


Figure 24. Equivalent Test Load

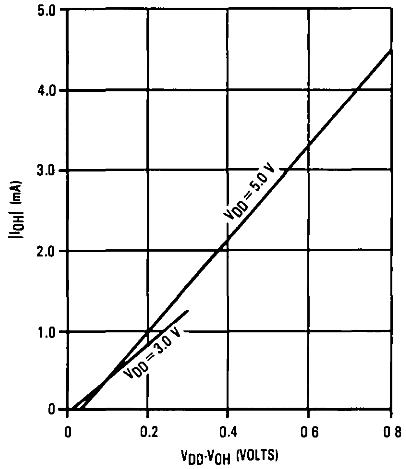


Figure 25. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

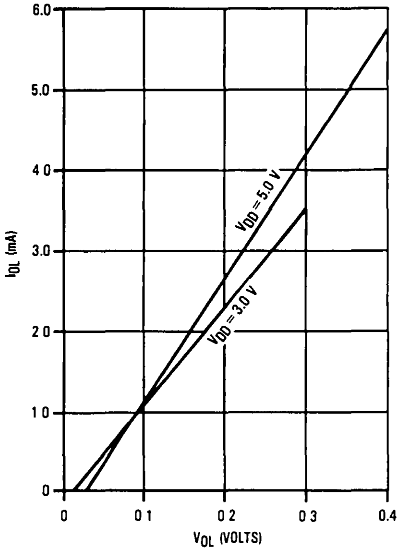


Figure 26. Typical V_{OL} vs I_{OL} for All Ports

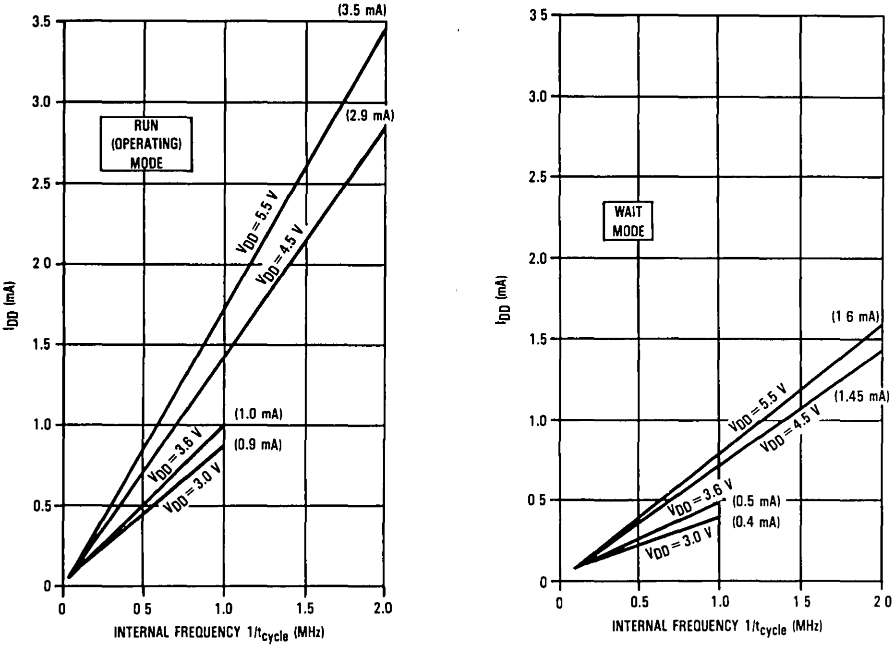
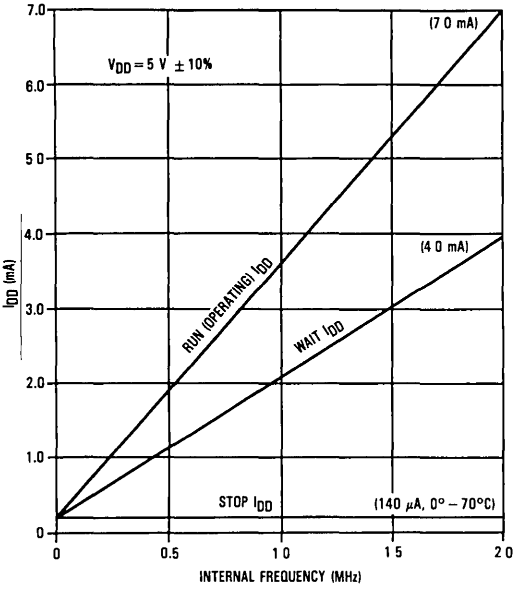


Figure 27. Typical Current vs Internal Frequency for Run and Wait Modes



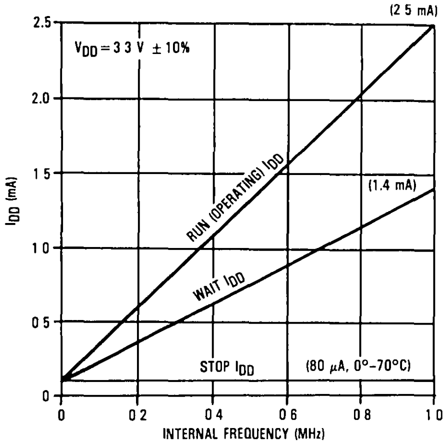


Figure 28. Maximum I_{DD} vs Frequency for $V_{DD}=5.0$ Vdc

Figure 29. Maximum I_{DD} vs Frequency for $V_{DD}=3.3$ Vdc

CONTROL TIMING

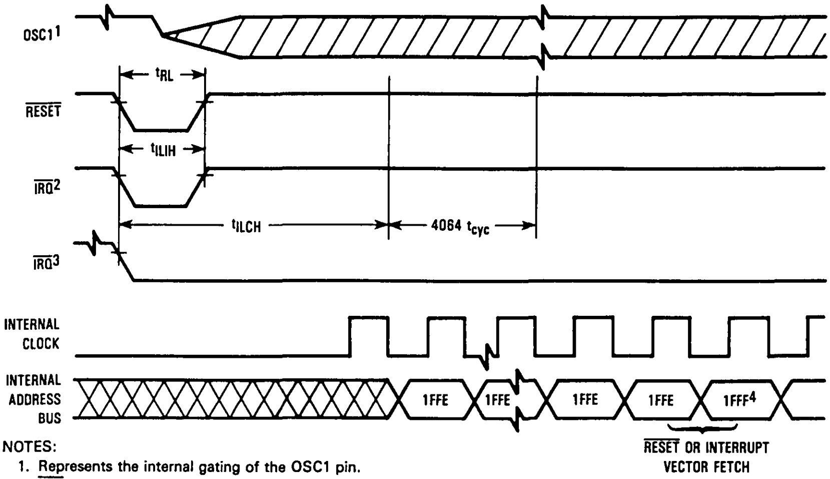
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} - 2$)		—	2.1	
External Clock ($f_{osc} - 2$)		dc	2.1	
Cycle Time (see Figure 33)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 33)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 30)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 33)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 31)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 31)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES:
1. Represents the internal gating of the OSC1 pin.
2. $\overline{\text{IRQ}}$ pin edge-sensitive mask option.
3. $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option.
4. $\overline{\text{RESET}}$ vector address shown for timing example.

Figure 30. Stop Recovery Timing Diagram

CONTROL TIMING

(V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} \div 2$)		—	1.0	
External Clock ($f_{osc} \div 2$)		dc	1.0	
Cycle Time (see Figure 33)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 33)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 30)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 33)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 31)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 31)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

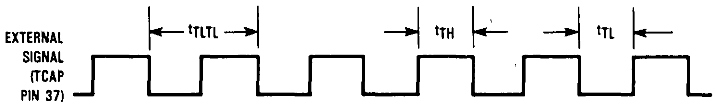


Figure 31. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	μs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	μs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

($V_{DD}=3.3$ Vdc ± 0.3 Vdc, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	µs
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	720	—	µs
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	720	—	µs
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	250	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	500	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, $C_L=200$ pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, $C_L=200$ pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

LCD DRIVER DC ELECTRICAL CHARACTERISTICS ($V_{GN}=0$)

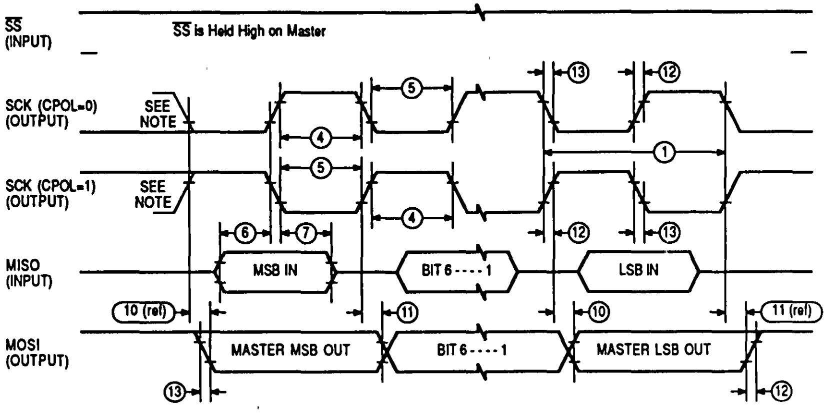
Characteristics	Symbol	V_{LL} V	−55°C		25°C			125°C		Unit
			Min	Max	Min	Typ	Max	Min	Max	
RMS Voltage Across a Segment "ON"	V_{ON}	3.0	—	—	—	1.73	—	—	—	V
(BPi-FPj) Segment		5.0	—	—	—	2.88	—	—	—	
"OFF"	V_{OFF}	3.0	—	—	—	1.00	—	—	—	V
Segment		5.0	—	—	—	1.67	—	—	—	
Average dc Offset Voltage	Vdc	3.0	—	30	—	10	30	—	30	mV
		5.0	—	50	—	17	50	—	50	
Output Current, Backplanes and Frontplanes										μA
High-Current State* $V_O=2.85$ V	I_{BH}	3.0	—	—	−100	−240	—	—	—	
$V_O=1.85$ V			—	—	−15	−35	—	—	—	
$V_O=1.15$ V			—	—	15	35	—	—	—	
$V_O=0.15$ V			—	—	100	240	—	—	—	
$V_O=4.85$ V		5.0	—	—	−200	−400	—	—	—	
$V_O=3.18$ V			—	—	−50	−110	—	—	—	
$V_O=1.82$ V			—	—	50	110	—	—	—	
$V_O=0.15$ V			—	—	200	400	—	—	—	
Low-Current State* $V_O=2.85$ V	I_{BL}	3.0	—	—	−100	−240	—	—	—	
$V_O=1.85$ V			—	—	−0.25	−1.0	—	—	—	
$V_O=1.15$ V			—	—	0.25	1.0	—	—	—	
$V_O=0.15$ V			—	—	100	240	—	—	—	
$V_O=4.85$ V		5.0	—	—	−200	−400	—	—	—	
$V_O=3.18$ V			—	—	−1.0	−3.0	—	—	—	
$V_O=1.82$ V			—	—	1.0	3.0	—	—	—	
$V_O=0.15$ V			—	—	200	400	—	—	—	

*For time $1/(256 \times f_{LCD})$ after the backplane or frontplane waveform changes to a new voltage level, the circuit is maintained in the high-current state to allow the load capacitors to charge quickly. Then the circuit is returned to the low-current state until the next voltage level change occurs.

LCD DRIVER DC ELECTRICAL CHARACTERISTICS ($V_{GN}=1$)

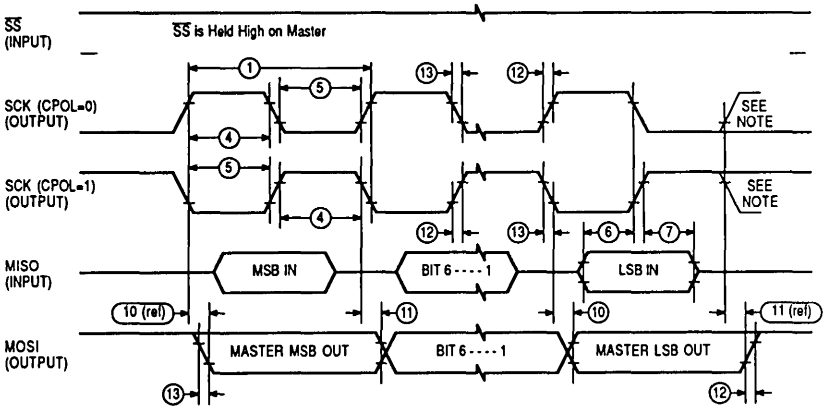
Characteristics	Symbol	V_{LL} V	−55°C		25°C			125°C		Unit
			Min	Max	Min	Typ	Max	Min	Max	
RMS Voltage Across a Segment "ON"	V_{ON}	3.0	—	—	—	1.73	—	—	—	V
(BPi-FPj) Segment		5.0	—	—	—	2.88	—	—	—	
"OFF"	V_{OFF}	3.0	—	—	—	1.00	—	—	—	V
Segment		5.0	—	—	—	1.67	—	—	—	
Average dc Offset Voltage	Vdc	3.0	—	30	—	10	30	—	30	mV
		5.0	—	50	—	17	50	—	50	
Output Current, Backplanes and Frontplanes										μA
High-Current State* $V_O=2.85$ V	I_{BH}	3.0	—	—	−100	−240	—	—	—	
$V_O=1.85$ V			—	—	−24	−75	—	—	—	
$V_O=1.15$ V			—	—	24	75	—	—	—	
$V_O=0.15$ V			—	—	100	240	—	—	—	
$V_O=4.85$ V		5.0	—	—	−200	−400	—	—	—	
$V_O=3.18$ V			—	—	−80	−250	—	—	—	
$V_O=1.82$ V			—	—	80	250	—	—	—	
$V_O=0.15$ V			—	—	200	400	—	—	—	
Low-Current State* $V_O=2.85$ V	I_{BL}	3.0	—	—	−100	−240	—	—	—	
$V_O=1.85$ V			—	—	−0.5	−2.0	—	—	—	
$V_O=1.15$ V			—	—	0.5	2.0	—	—	—	
$V_O=0.15$ V			—	—	100	240	—	—	—	
$V_O=4.85$ V		5.0	—	—	−200	−400	—	—	—	
$V_O=3.18$ V			—	—	−2.0	−7.0	—	—	—	
$V_O=1.82$ V			—	—	2.0	7.0	—	—	—	
$V_O=0.15$ V			—	—	200	400	—	—	—	

*For time $1/(256 \times f_{LCD})$ after the backplane or frontplane waveform changes to a new voltage level, the circuit is maintained in the high-current state to allow the load capacitors to charge quickly. Then the circuit is returned to the low-current state until the next voltage level change occurs.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

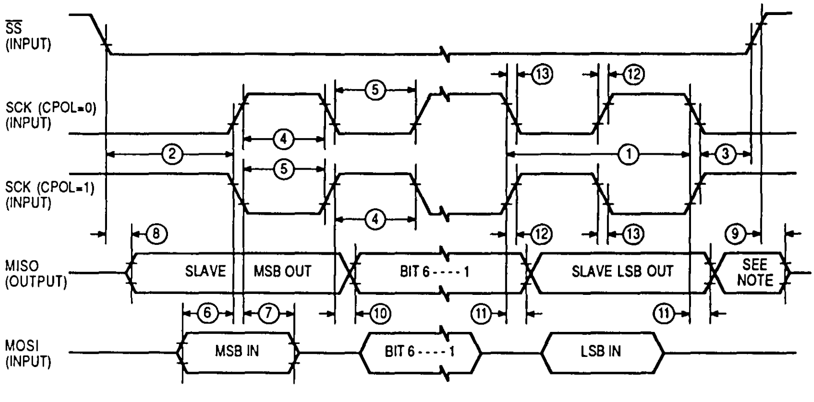
a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

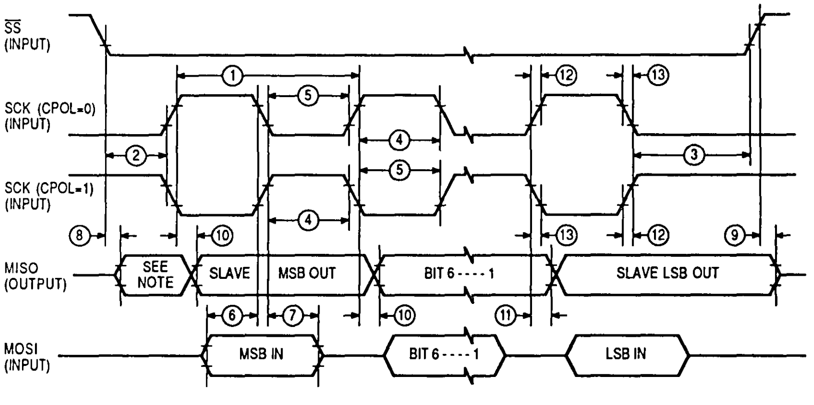
b) SPI MASTER TIMING (CPHA=1)

Figure 32. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally MSB of character just received

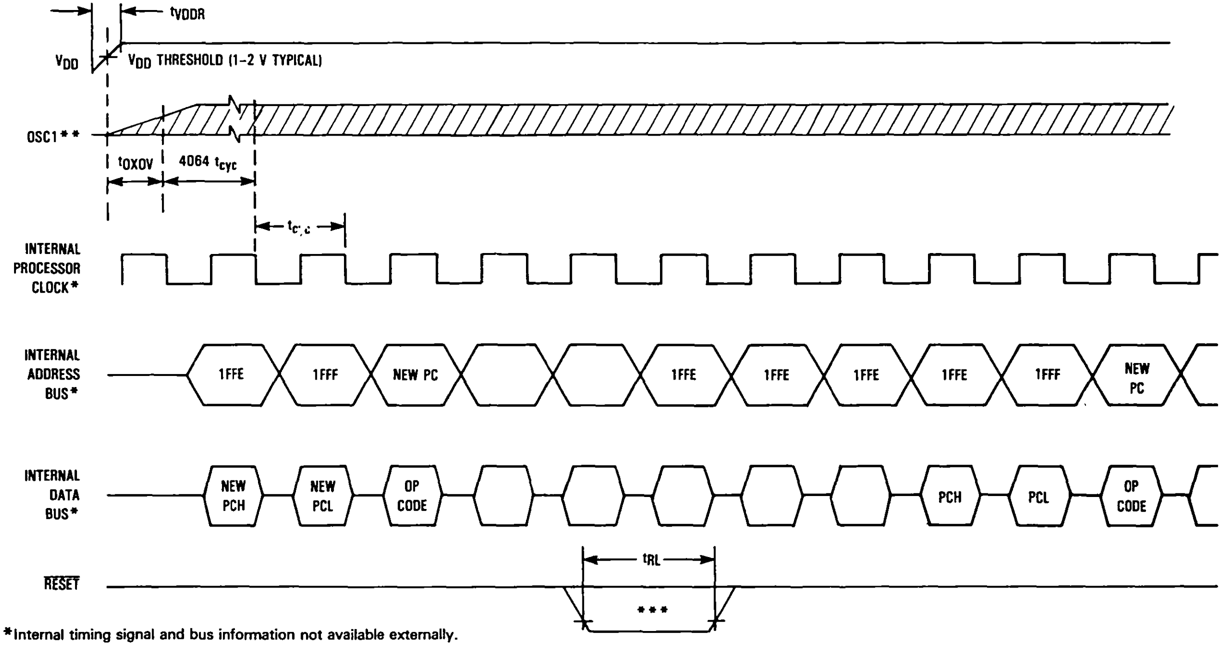
c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA = 1)

Figure 32. SPI Timing Diagrams (Sheet 2 of 2)



tVDDR
VDD
VDD THRESHOLD (1-2 V TYPICAL)
OSC1**
tOXOV
4064 tcyc
tcyc
INTERNAL PROCESSOR CLOCK*
INTERNAL ADDRESS BUS*
1FFE
1FFF
NEW PC
1FFE
1FFE
1FFE
1FFE
1FFF
NEW PC
INTERNAL DATA BUS*
NEW PCH
NEW PCL
OP CODE
PCH
PCL
OP CODE
RESET
tRL
***

*Internal timing signal and bus information not available externally.

**OSC1 line is not meant to represent frequency. It is only used to represent time.

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 33. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS™, disk file
MS™-DOS/PC-DOS disk file (360K)
EPROM MCM68766

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

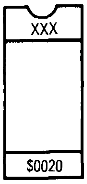
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

Start the page zero, user ROM at EPROM address $0020 through $004F. Start the user ROM at EPROM address $0100 through $19FF with vectors from $1FF4 to $1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05L6 device.

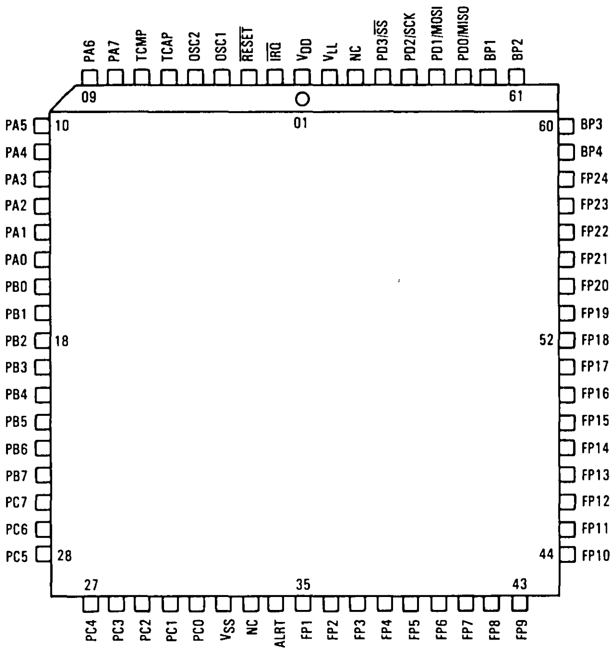
Package Type	Temperature	MC Order Number
PLCC[1] (FN Suffix)	0°C to +70°C	MC68HC05L6FN

1. PLCC packaging is available for samples and evaluation module support only.

MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

68-PIN PLCC PACKAGE


PA6
PA7
TCMP
TCAP
OSC2
OSC1
RESET
IRQ
V_{DD}
V_{LL}
NC
PD3/SS
PD2/SCK
PD1/MOSI
PD0/MISO
BP1
BP2
09
61
01
10
60
PA5
PA4
PA3
PA2
PA1
PA0
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC7
PC6
PC5
18
28
52
44
BP3
BP4
FP24
FP23
FP22
FP21
FP20
FP19
FP18
FP17
FP16
FP15
FP14
FP13
FP12
FP11
FP10
27
35
43
PC4
PC3
PC2
PC1
PC0
V_{SS}
NC
ALRT
FP1
FP2
FP3
FP4
FP5
FP6
FP7
FP8
FP9

MC68HC05M4

Product Preview

8-Bit Microcontroller Unit

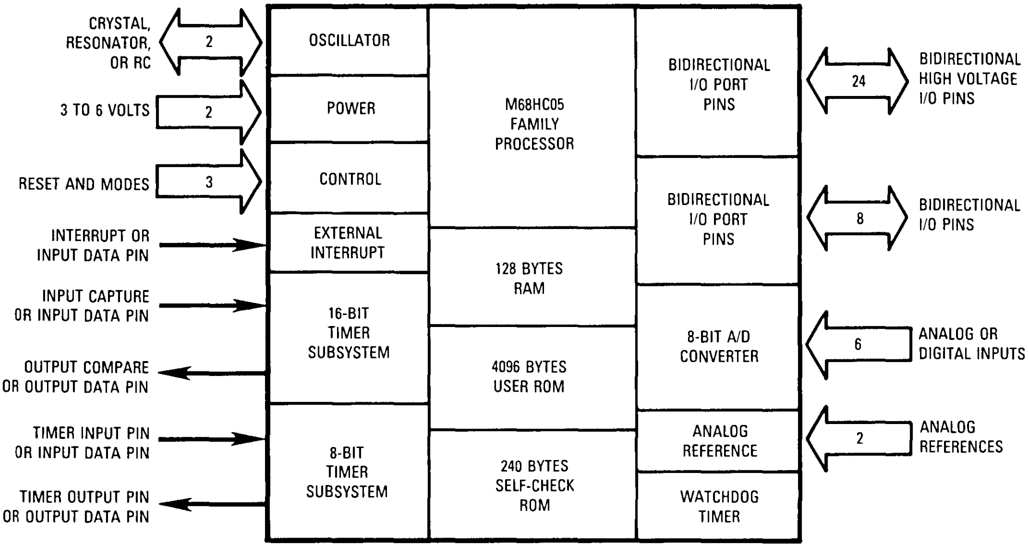
The MC68HC05M4 is an advanced 8-bit microcontroller unit (MCU) with highly sophisticated on-chip peripheral capabilities. This device contains 4096 bytes of ROM, a 16-bit free-running timer subsystem and a 8-bit countdown timer subsystem. In addition, 24 of the I/O lines can drive vacuum fluorescent (VF) grids and anodes directly.

The following are some of the features of the MC68HC05M4.

- HCMOS Technology
- Fully Static Operation
- 4096 Bytes of User ROM
- 128 Bytes of RAM
- 240 Bytes of Self-Check ROM
- 32 Bidirectional I/O Lines
- 24 I/O Lines Mask Selectable as High Voltage VF Drivers
- 16-Bit Timer Subsystem
- 8-Bit Timer Subsystem
- 8-Bit A/D Converter
- 6 Analog Inputs
- Self-Check Mode
- Interrupts: External, 16-Bit Timer, and 8-Bit Timer
- Watchdog System
- Single 3- to 6-Volt Supply
- 2.1 MHz Internal Frequency at 5 Volts; 1.0 MHz at 3 Volts
- Power Saving STOP, WAIT, and Data Retention Modes
- Versatile Interrupt Handling
- True Bit Manipulation Set
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Multiply Instruction
- 52-Pin PLCC Package

This document contains information on a product under development Motorola reserves the right to change or discontinue this product without notice

BLOCK DIAGRAM


CRYSTAL, RESONATOR, OR RC
2
3 TO 6 VOLTS
2
RESET AND MODES
3
INTERRUPT OR INPUT DATA PIN
INPUT CAPTURE OR INPUT DATA PIN
OUTPUT COMPARE OR OUTPUT DATA PIN
TIMER INPUT PIN OR INPUT DATA PIN
TIMER OUTPUT PIN OR OUTPUT DATA PIN
OSCILLATOR
POWER
CONTROL
EXTERNAL INTERRUPT
16-BIT TIMER SUBSYSTEM
8-BIT TIMER SUBSYSTEM
M68HC05 FAMILY PROCESSOR
128 BYTES RAM
4096 BYTES USER ROM
240 BYTES SELF-CHECK ROM
BIDIRECTIONAL I/O PORT PINS
BIDIRECTIONAL I/O PORT PINS
8-BIT A/D CONVERTER
ANALOG REFERENCE
WATCHDOG TIMER
24
BIDIRECTIONAL HIGH VOLTAGE I/O PINS
8
BIDIRECTIONAL I/O PINS
6
ANALOG OR DIGITAL INPUTS
2
ANALOG REFERENCES

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC05P1

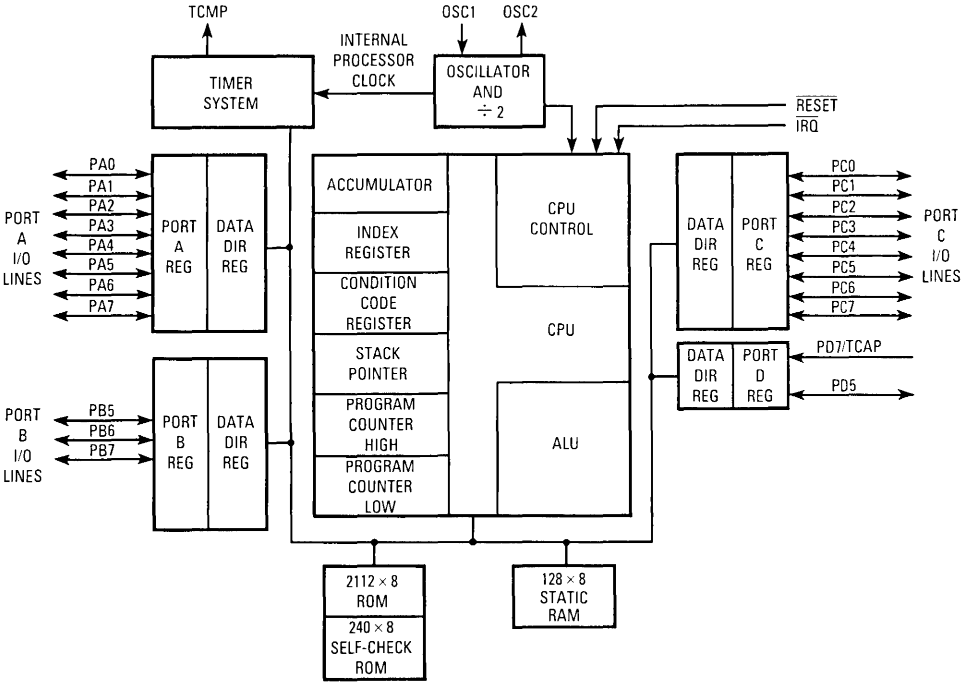
Technical Summary
8-Bit Microcontroller Unit

The MC68HC05P1 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-cost MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 128 Bytes of On-Chip RAM
- 2112 Bytes of User ROM
- 20 Bidirectional I/O Lines and One Input-Only Line
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8×8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
INTERNAL PROCESSOR CLOCK
OSC1
OSC2
TIMER SYSTEM
OSCILLATOR AND ÷ 2
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A I/O LINES
PORT A REG
DATA DIR REG
ACCUMULATOR
INDEX REGISTER
CONDITION CODE REGISTER
STACK POINTER
PROGRAM COUNTER HIGH
PROGRAM COUNTER LOW
CPU CONTROL
CPU
ALU
DATA DIR REG
PORT C REG
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PORT C I/O LINES
DATA DIR REG
PORT D REG
PD7/TCAP
PD5
PORT B I/O LINES
PB5
PB6
PB7
PORT B REG
DATA DIR REG
2112 × 8 ROM
240 × 8 SELF-CHECK ROM
128 × 8 STATIC RAM

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins, providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

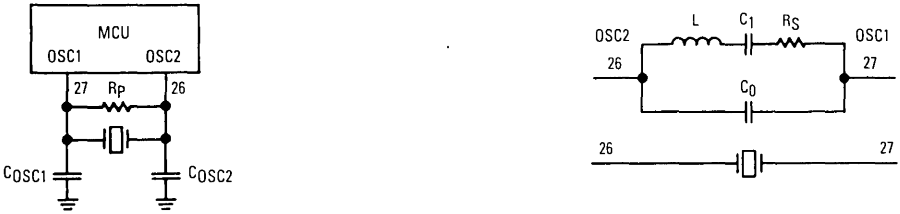
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15–40	15–30	pF
C_{OSC2}	15–30	15–25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

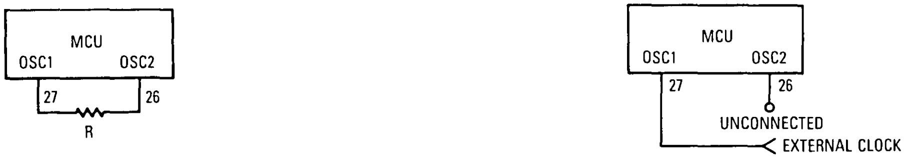
	2–4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1–10	MΩ
Q	1250	—

(a) Crystal Ceramic Resonator Parameters



(b) Crystal Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

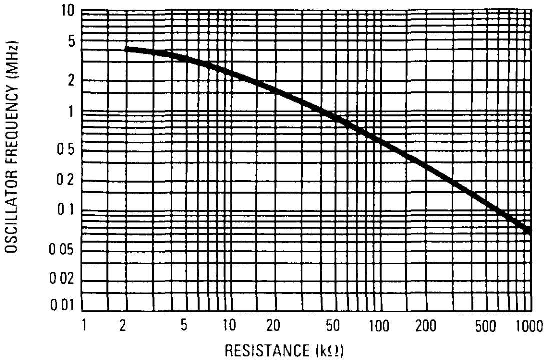


Figure 2. Typical Frequency vs. Resistance for RC Oscillator Option Only (Accuracy = ±50%)

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA7–PA0, PB7–PB5, PC7–PC0)

These 19 lines are arranged in two 8-bit ports and one 3-bit port (A, C, and B). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

SPECIAL-FUNCTION PORT (PD5–PD7)

These three lines comprise port D, a special-function port. Bit 7 is an input-only pin that is shared with TCAP (refer to **TIMER** for information on TCAP). Bit 6 is an output-only pin that serves as TCMP (refer to **TIMER** for more information on TCMP). Bit 5 is a normal I/O pin. All other bits are not implemented and read as zeros.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

SPECIAL-FUNCTION PORT PROGRAMMING

Port D is a special-function port consisting of three bits, PD7–PD5. Bit 7 is an input-only bit shared with TCAP. This bit can be read at any time, even if the TCAP function is enabled. Bit 6 is an output-only pin that serves as TCMP.

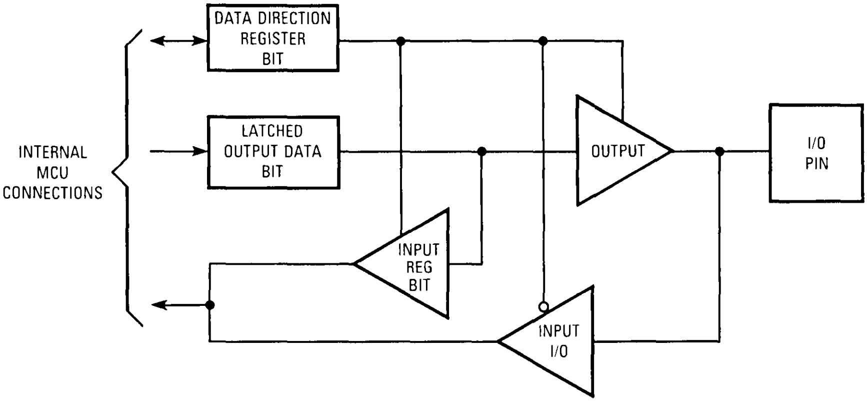


Figure 3. Typical Port I/O Circuit

Table 1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode The output data latch is read.

*R/$\overline{W}$ is an internal signal.

It cannot be used as a normal output, only as TCMP, and always reads as zero. Bit 5 is a normal I/O pin with a corresponding DDR bit at address $0007. All other DDR bits are not implemented and read as zeros. Bits 4–0 of the port D data register are not implemented. Bit 4 always reads as one; the other bits always read as zeros.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF0–$1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

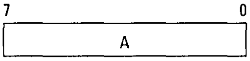
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

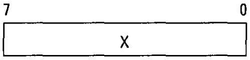
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



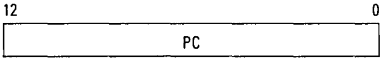
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are

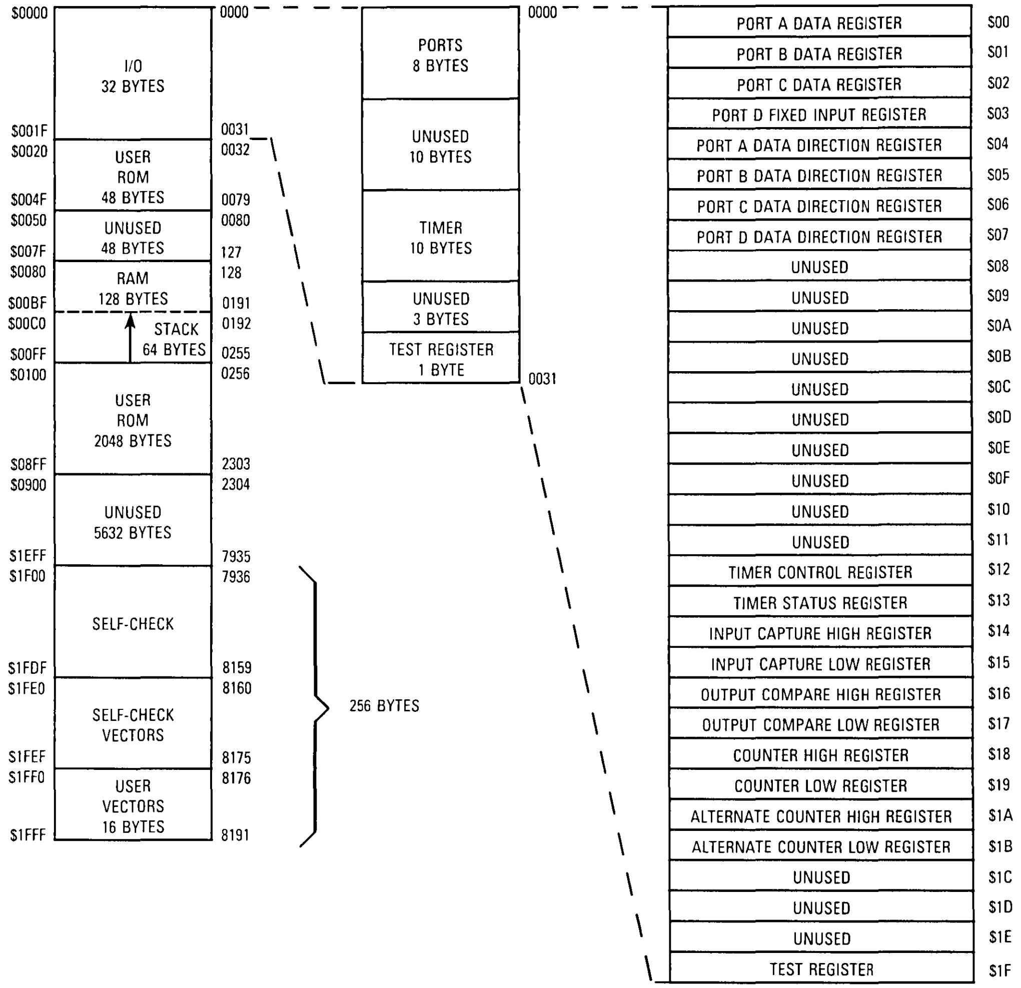
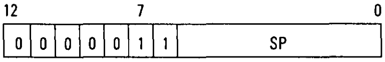

$0000
I/O
32 BYTES
$001F
$0020
USER
ROM
48 BYTES
$004F
$0050
UNUSED
48 BYTES
$007F
$0080
RAM
128 BYTES
$00BF
$00C0
STACK
64 BYTES
$00FF
$0100
USER
ROM
2048 BYTES
$08FF
$0900
UNUSED
5632 BYTES
$1EFF
$1F00
SELF-CHECK
$1FDF
$1FE0
SELF-CHECK
VECTORS
$1FEF
$1FF0
USER
VECTORS
16 BYTES
$1FFF
0000
0031
0032
0079
0080
127
128
0191
0192
0255
0256
2303
2304
7935
7936
8159
8160
8175
8176
8191
256 BYTES
0000
PORTS
8 BYTES
UNUSED
10 BYTES
TIMER
10 BYTES
UNUSED
3 BYTES
TEST REGISTER
1 BYTE
0031
0000
PORT A DATA REGISTER
$00
PORT B DATA REGISTER
$01
PORT C DATA REGISTER
$02
PORT D FIXED INPUT REGISTER
$03
PORT A DATA DIRECTION REGISTER
$04
PORT B DATA DIRECTION REGISTER
$05
PORT C DATA DIRECTION REGISTER
$06
PORT D DATA DIRECTION REGISTER
$07
UNUSED
$08
UNUSED
$09
UNUSED
$0A
UNUSED
$0B
UNUSED
$0C
UNUSED
$0D
UNUSED
$0E
UNUSED
$0F
UNUSED
$10
UNUSED
$11
TIMER CONTROL REGISTER
$12
TIMER STATUS REGISTER
$13
INPUT CAPTURE HIGH REGISTER
$14
INPUT CAPTURE LOW REGISTER
$15
OUTPUT COMPARE HIGH REGISTER
$16
OUTPUT COMPARE LOW REGISTER
$17
COUNTER HIGH REGISTER
$18
COUNTER LOW REGISTER
$19
ALTERNATE COUNTER HIGH REGISTER
$1A
ALTERNATE COUNTER LOW REGISTER
$1B
UNUSED
$1C
UNUSED
$1D
UNUSED
$1E
TEST REGISTER
$1F

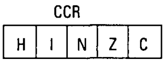
Figure 4. Memory Map

appended to the six least significant register bits to produce an address within the range of $00FF–$00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.


12
7
0
0
0
0
0
0
1
1
SP

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

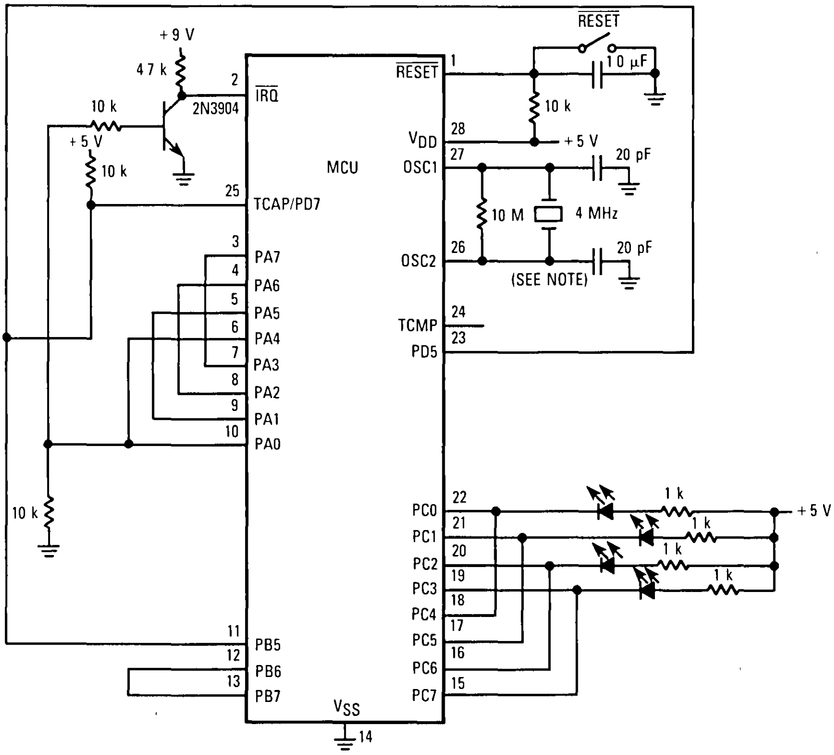
The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC3–PC0 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF bit
Interrupts — Tests external and timer interrupts

Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter


+9 V
47 k
2
IRQ
10 k
2N3904
+5 V
10 k
25
TCAP/PD7
MCU
3
PA7
4
PA6
5
PA5
6
PA4
7
PA3
8
PA2
9
PA1
10
PA0
10 k
11
PB5
12
PB6
13
PB7
VSS
14
RESET
1
RESET
10 µF
10 k
28
VDD
+5 V
27
OSC1
20 pF
10 M
4 MHz
26
OSC2
20 pF
(SEE NOTE)
24
TCMP
23
PD5
22
PC0
1 k
+5 V
21
PC1
1 k
20
PC2
1 k
19
PC3
1 k
18
PC4
17
PC5
16
PC6
15
PC7

NOTE The RC Oscillator option can also be used in this circuit.

Figure 5. Self-Check Circuit Schematic Diagram

until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations \$0080–\$0081 are overwritten. Upon return to the user's program, X = 40. If the test passed, A = 0.

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad ROM
1	1	0	1	Bad Interrupts or $\overline{\text{IRQ}}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on; 1 indicates LED is off.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM checksum subroutine is called at location \$1F8A with RAM location \$0083 equal to \$01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations \$0080 through \$0083 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 4064 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted three different ways: the two maskable hardware interrupts ($\overline{\text{IRQ}}$ and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

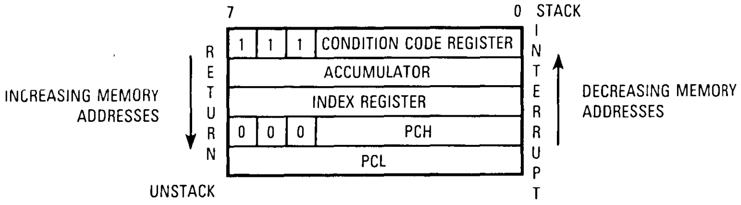
The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and



NOTE Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 6. Interrupt Stacking Order

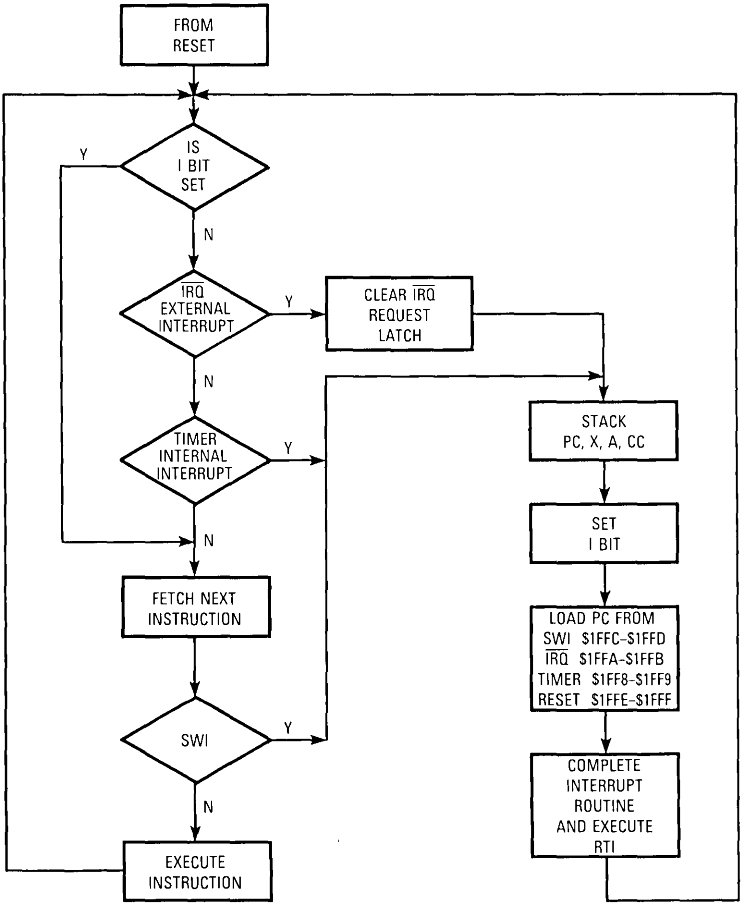

FROM RESET
IS I BIT SET
Y
N
IRQ EXTERNAL INTERRUPT
Y
CLEAR IRQ REQUEST LATCH
N
TIMER INTERNAL INTERRUPT
Y
N
FETCH NEXT INSTRUCTION
SWI
Y
N
EXECUTE INSTRUCTION
STACK PC, X, A, CC
SET I BIT
LOAD PC FROM
SWI $1FFC-$1FFD
IRQ $1FFA-$1FFB
TIMER $1FF8-$1FF9
RESET $1FFE-$1FFF
COMPLETE INTERRUPT ROUTINE AND EXECUTE RTI

Figure 7. Reset and Interrupt Processing Flowchart

the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

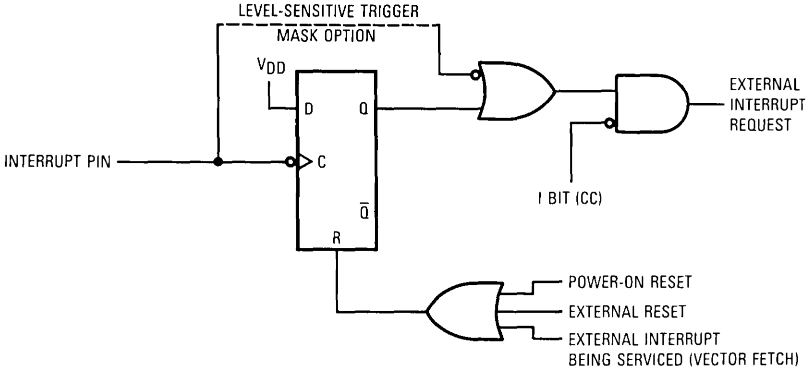
EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt, except that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

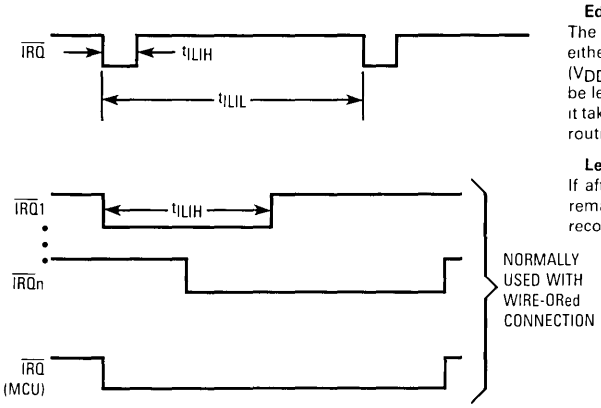
Either a level-sensitive and edge-sensitive trigger or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.


LEVEL-SENSITIVE TRIGGER
MASK OPTION
V_{DD}
D
Q
C
$\overline{Q}$
R
INTERRUPT PIN
EXTERNAL
INTERRUPT
REQUEST
I BIT (CC)
POWER-ON RESET
EXTERNAL RESET
EXTERNAL INTERRUPT
BEING SERVICED (VECTOR FETCH)

(a) Interrupt Internal Function Diagram


$\overline{IRQ}$
t_{ILIH}
t_{ILIL}
$\overline{IRQ}1$
t_{ILIH}
$\overline{IRQ}n$
$\overline{IRQ}$
(MCU)
NORMALLY
USED WITH
WIRE-ORed
CONNECTION

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V) The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 8. External Interrupt

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer remains active (refer to Figure 10). An interrupt from the timer can cause the MCU to exit the WAIT mode.

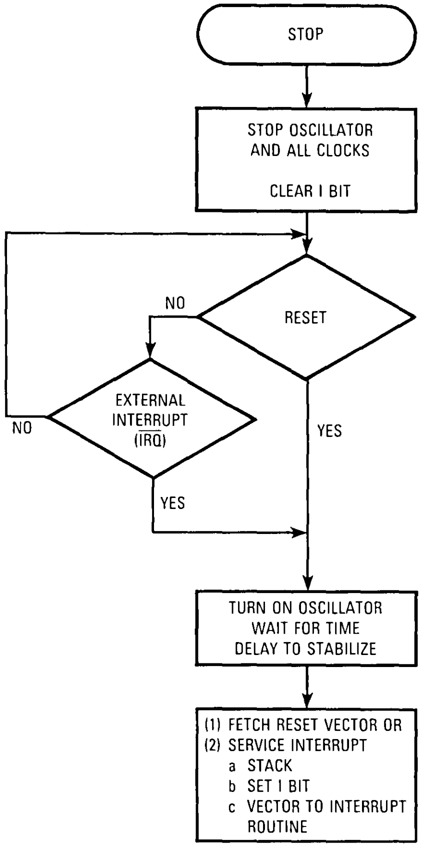

STOP
STOP OSCILLATOR AND ALL CLOCKS
CLEAR I BIT
NO
RESET
YES
EXTERNAL INTERRUPT ($\overline{IRQ}$)
NO
YES
TURN ON OSCILLATOR WAIT FOR TIME DELAY TO STABILIZE
(1) FETCH RESET VECTOR OR
(2) SERVICE INTERRUPT
a STACK
b SET I BIT
c VECTOR TO INTERRUPT ROUTINE

Figure 9. STOP Function Flowchart

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{\text{RESET}}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, \$18–\$19 (counter register) or \$1A–\$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter (\$19, \$1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) (\$18, \$1A), the LSB (\$19, \$1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB (\$19 or \$1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register LSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to \$FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to \$FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from \$FFFF to \$0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

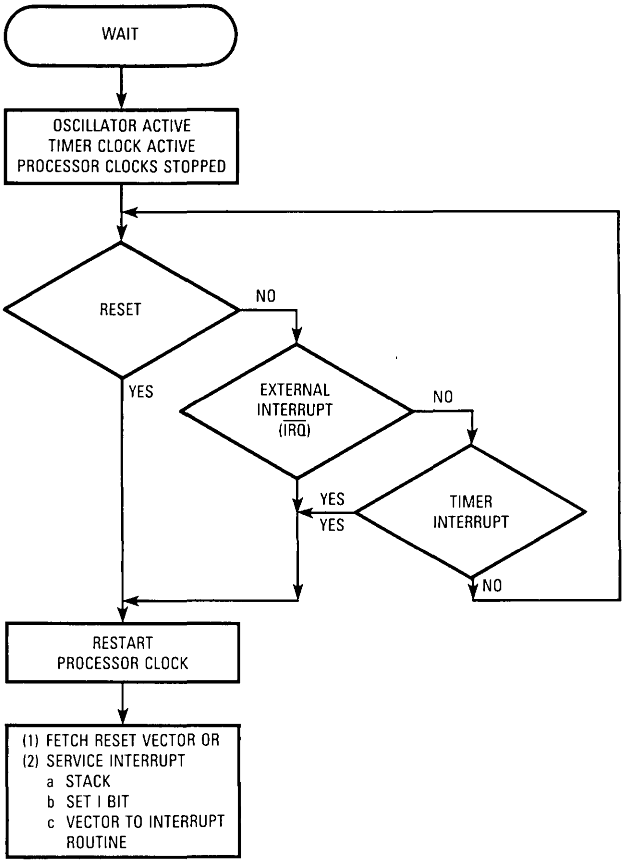

WAIT
OSCILLATOR ACTIVE
TIMER CLOCK ACTIVE
PROCESSOR CLOCKS STOPPED
RESET
NO
YES
EXTERNAL INTERRUPT (IRQ)
NO
YES
YES
TIMER INTERRUPT
NO
RESTART PROCESSOR CLOCK
(1) FETCH RESET VECTOR OR
(2) SERVICE INTERRUPT
a STACK
b SET I BIT
c VECTOR TO INTERRUPT ROUTINE

Figure 10. WAIT Function Flowchart

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and, if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register, regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the

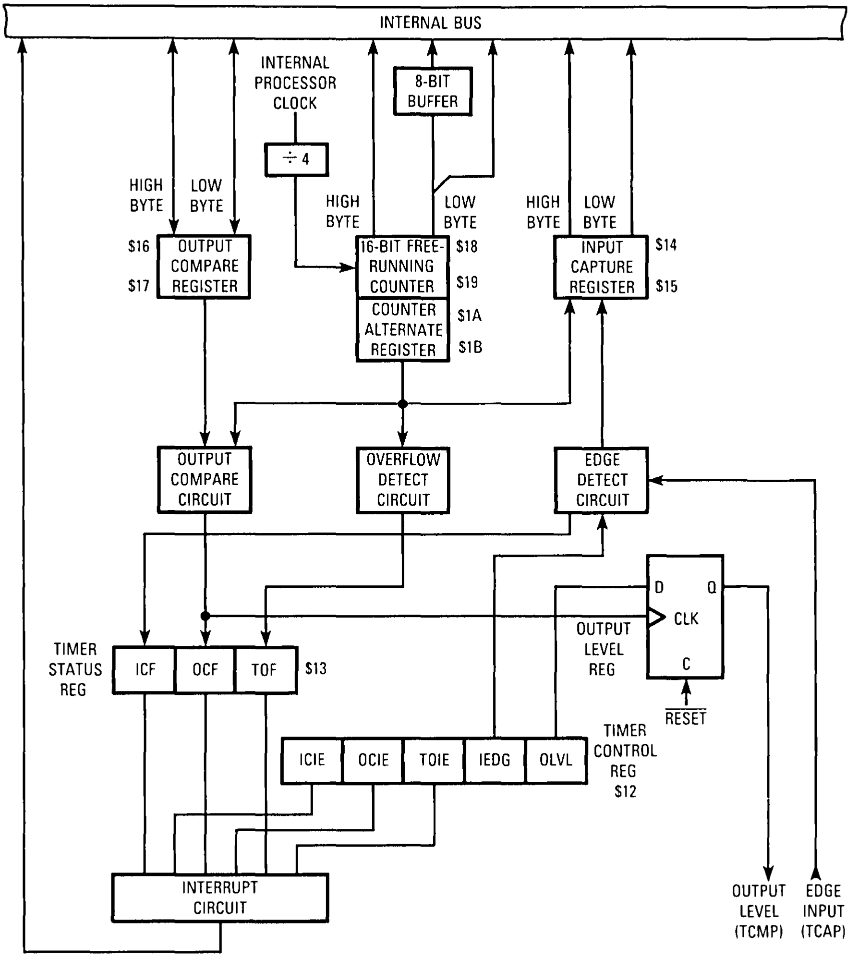

INTERNAL BUS
INTERNAL PROCESSOR CLOCK
÷ 4
8-BIT BUFFER
HIGH BYTE
LOW BYTE
HIGH BYTE
LOW BYTE
HIGH BYTE
LOW BYTE
$16
$17
OUTPUT COMPARE REGISTER
16-BIT FREE-RUNNING COUNTER
$18
$19
COUNTER ALTERNATE REGISTER
$1A
$1B
INPUT CAPTURE REGISTER
$14
$15
OUTPUT COMPARE CIRCUIT
OVERFLOW DETECT CIRCUIT
EDGE DETECT CIRCUIT
D
Q
CLK
C
OUTPUT LEVEL REG
RESET
TIMER STATUS REG
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
TIMER CONTROL REG
$12
INTERRUPT CIRCUIT
OUTPUT LEVEL (TCMP)
EDGE INPUT (TCAP)

Figure 11. Timer Block Diagram

value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET.							
0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

IEDG — Input Edge

Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
- 1 = Positive edge
- 0 = Negative edge

Reset does not affect the IEDG bit (U = unaffected).

OLVL — Output Level

Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
- 1 = High output
- 0 = Low output

Bits 2, 3, and 4 — Not used

Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

7	6	5	4	3	2	1	0
ICF	OCF	TOF	0	0	0	0	0
RESET							
U	U	U	0	0	0	0	0

ICF — Input Capture Flag
- 1 = Flag set when selected polarity edge is sensed by input capture edge detector
- 0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF — Output Compare Flag
- 1 = Flag set when output compare register contents match the free-running counter contents
- 0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
- 1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
- 0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used

Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A ← X × A		
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register		
Condition Codes	H: Cleared I Not affected N: Not affected Z Not affected C: Cleared		
Source Form(s)	MUL		
Addressing Mode	Cycles	Bytes	Opcode
Inherent	11	1	$42

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . 7)
Set Bit n	BSET n (n=0 . . 7)
Clear Bit n	BCLR n (n=0 . 7)

OPCODE MAP SUMMARY

Table 3 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in

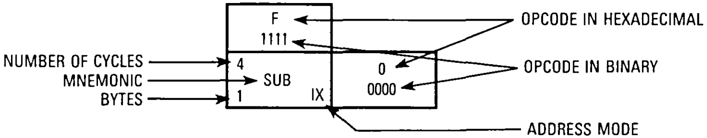
Table 3. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
HI LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI LOW
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 2 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 3 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

Legend



which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

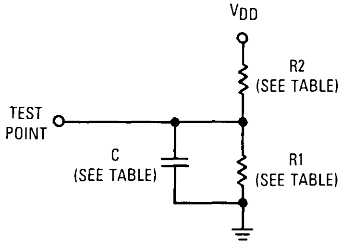
MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to 2×V_{DD}+0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05P1P, DW	T_A	T_L to T_H 0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		60	
SOIC		60	



V_{DD}=4.5 V

Pins	R1	R2	C
PA7–PA0, PB7–PB5, PC7–PC0 PD5, TCMP	3.26 kΩ	2.38 kΩ	50 pF

V_{DD}=3.0 V

Pins	R1	R2	C
PA7–PA0, PB7–PB5, PC7–PC0 PD5, TCMP	10.91 kΩ	6.32 kΩ	50 pF

Figure 12. Equivalent Test Load

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS

($V_{DD} = 5.0$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (see Figure 13) ($I_{Load} = 0.8$ mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OH}	$V_{DD} - 0.8$	—	—	V
Output Low Voltage (see Figure 14) ($I_{Load} = 1.6$ mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 15 and 16)		—	3.5	7.0	mA
Wait (see Figures 15 and 16)		—	1.6	4.0	mA
Stop (see Figure 16)					
25°C		—	2.0	50	µA
0° to 70°C (Standard)		—	—	140	µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB5, PC7–PC0, PD5	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1, PD5, PD7/TCAP	I_{in}	—	—	±1	µA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{RESET}$, $\overline{IRQ}$, PD5, PD7/TCAP	C_{in}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source ($f_{osc} = 4.2$ MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, $C_L = 20$ pF on OSC2.
4. Wait, Stop I_{DD}: All ports configured as inputs, $V_{IL} = 0.2$ V, $V_{IH} = V_{DD} - 0.2$ V.
5. Stop I_{DD} measured with OSC1 = V_{SS}.
6. Standard temperature range is 0° to 70°C.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \le 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (see Figure 13) (I_{Load} = 0.2 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OH}	$V_{DD} - 0.3$	—	—	V
Output Low Voltage (see Figure 14) (I_{Load} = 0.4 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 15 and 17)		—	1.0	2.5	mA
Wait (see Figures 15 and 17)		—	0.5	1.4	mA
Stop (see Figure 17)					
25°C		—	1.0	30	µA
0° to 70°C (Standard)		—	—	80	µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB5, PC7–PC0, PD5	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1, PD5, PD7/TCAP	I_{in}	—	—	±1	µA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, PD5, PD7/TCAP	C_{in}	—	—	8	

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Run (Operating) I_{DD}, Wait I_{DD} Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
4. Wait, Stop I_{DD} All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
5. Stop I_{DD} measured with OSC1 = V_{SS}.
6. Standard temperature range is 0° to 70°C.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ μA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (see Figure 13) (I_{Load} = 0.8 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OH}	$V_{DD} - 0.8$	—	—	V
Output Low Voltage (see Figure 14) (I_{Load} = 1.6 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 15 and 16)		—	3.5	7.0	mA
Wait (see Figures 15 and 16)		—	1.6	4.0	mA
Stop (see Figure 16)					
25°C		—	2.0	50	μA
0° to 70°C (Standard)		—	—	140	μA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB5, PC7–PC0, PD5	I_{IL}	—	—	± 10	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1, PD5, PD7/TCAP	I_{in}	—	—	± 1	μA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, PD5, PD7/TCAP	C_{in}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only
3. Run (Operating) I_{DD}, Wait I_{DD}. Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
4. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
5. Stop I_{DD} measured with OSC1 = V_{SS}.
6. Standard temperature range is 0° to 70°C.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

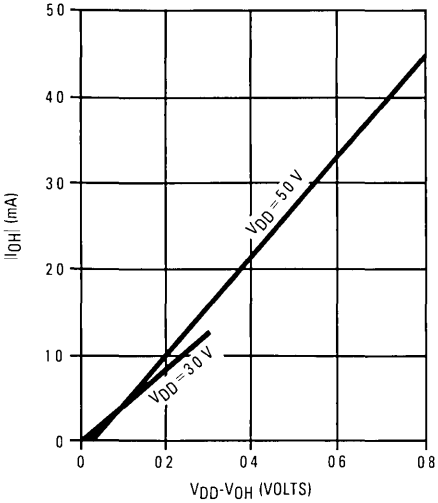


Figure 13. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

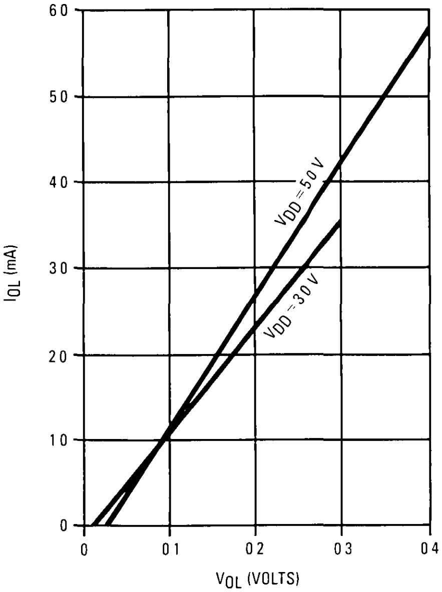


Figure 14. Typical V_{OL} vs I_{OL} for All Ports

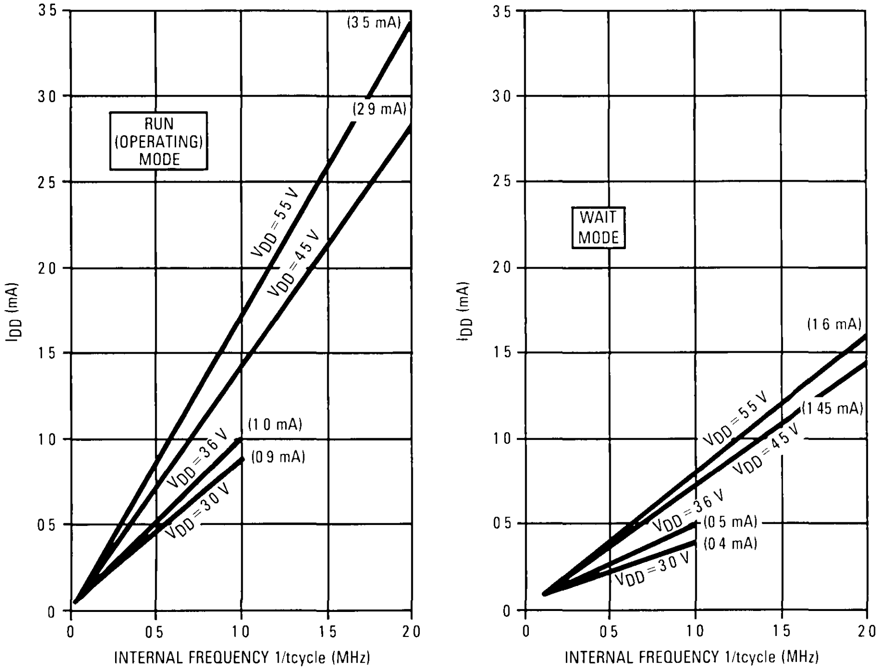


Figure 15. Typical Current vs Internal Frequency for Run and Wait Modes

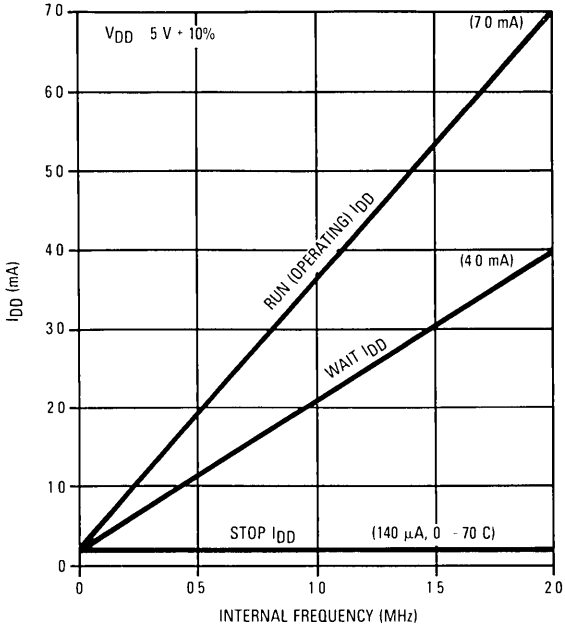


Figure 16. Maximum I_{DD} vs Frequency for V_{DD} = 5.0 Vdc

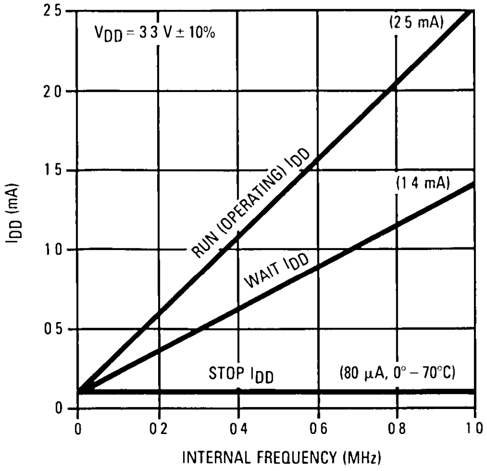


Figure 17. Maximum I_{DD} vs Frequency for V_{DD} = 3.3 Vdc

CONTROL TIMING

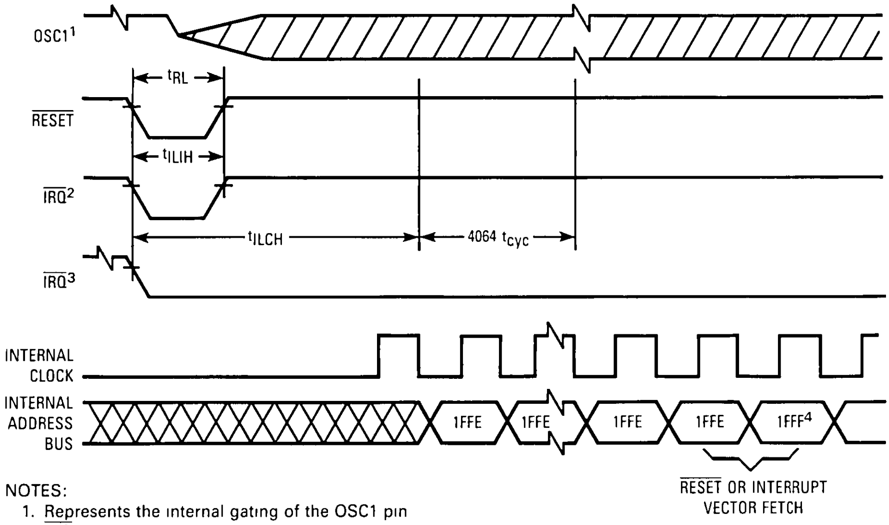
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal (f_{osc} − 2)		—	2.1	
External Clock (f_{osc} − 2)		dc	2.1	
Cycle Time (see Figure 20)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 20)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 18)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 20)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 19)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 19)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES:
1. Represents the internal gating of the OSC1 pin
2. $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3. $\overline{\text{IRQ}}$ pin level- and edge-sensitive mask option.
4. $\overline{\text{RESET}}$ vector address shown for timing example

Figure 18. Stop Recovery Timing Diagram

CONTROL TIMING

($V_{DD} = 3.3$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} - 2$)		—	1.0	
External Clock ($f_{osc} - 2$)		dc	1.0	
Cycle Time (see Figure 20)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 20)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 18)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 20)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 19)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 19)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

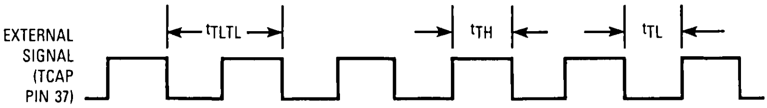
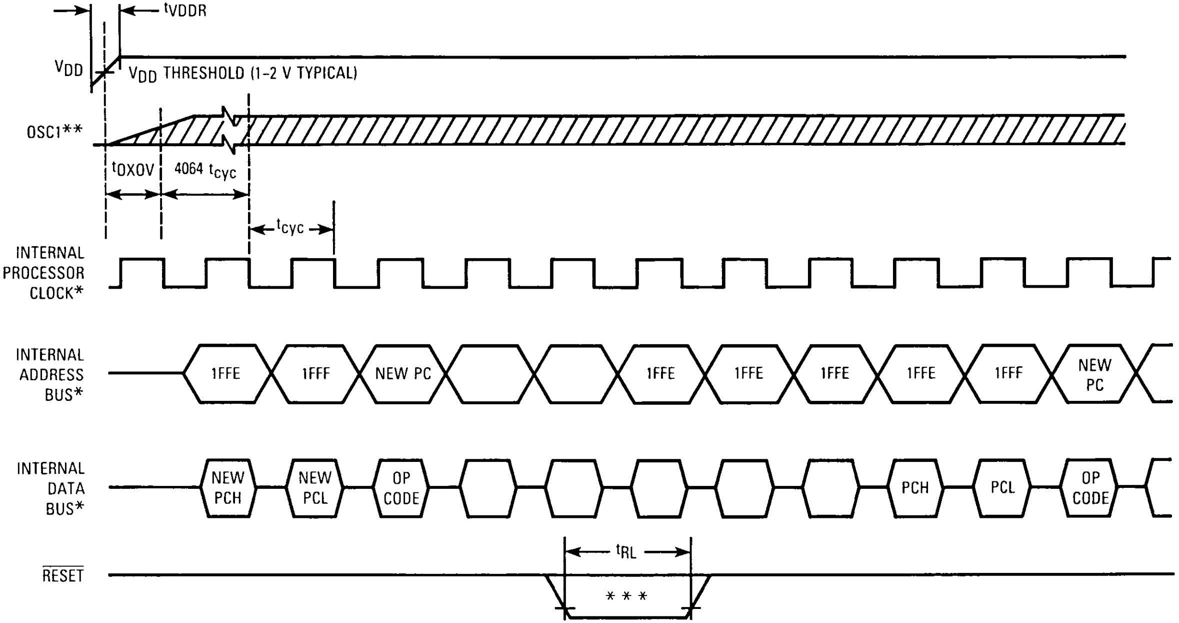


Figure 19. Timer Relationships


tVDDR
VDD
VDD THRESHOLD (1-2 V TYPICAL)
OSC1**
tOXOV
4064 tcyc
tcyc
INTERNAL PROCESSOR CLOCK*
INTERNAL ADDRESS BUS*
1FFE
1FFF
NEW PC
1FFE
1FFE
1FFE
1FFE
1FFF
NEW PC
INTERNAL DATA BUS*
NEW PCH
NEW PCL
OP CODE
PCH
PCL
OP CODE
RESET
tRL
* * *

*Internal timing signal and bus information not available externally.

**OSC1 line is not meant to represent frequency It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of RESET initiates the reset sequence

Figure 20. Power-On Reset and RESET

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS®, disk file
MS®-DOS/PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM MC68HC805C4

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

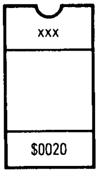
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2764, 68764, or 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

For an MC68HC805C4 MCU start the page zero, user ROM at EEPROM address $0020 through $004F. Start the user ROM at EEPROM address $0100 through $08FF with vectors from $1FF0 to $1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



XXX = CUSTOMER ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

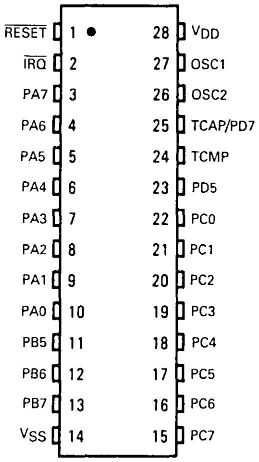
The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05P1 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C	MC68HC05P1P
SOIC (DW Suffix)	0°C to +70°C	MC68HC05P1DW

MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

28-PIN DUAL-IN-LINE PACKAGE AND
28-PIN SOIC PACKAGE


RESET 1
IRQ 2
PA7 3
PA6 4
PA5 5
PA4 6
PA3 7
PA2 8
PA1 9
PA0 10
PB5 11
PB6 12
PB7 13
VSS 14
28 VDD
27 OSC1
26 OSC2
25 TCAP/PD7
24 TCMP
23 PD5
22 PC0
21 PC1
20 PC2
19 PC3
18 PC4
17 PC5
16 PC6
15 PC7

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HCL05C4

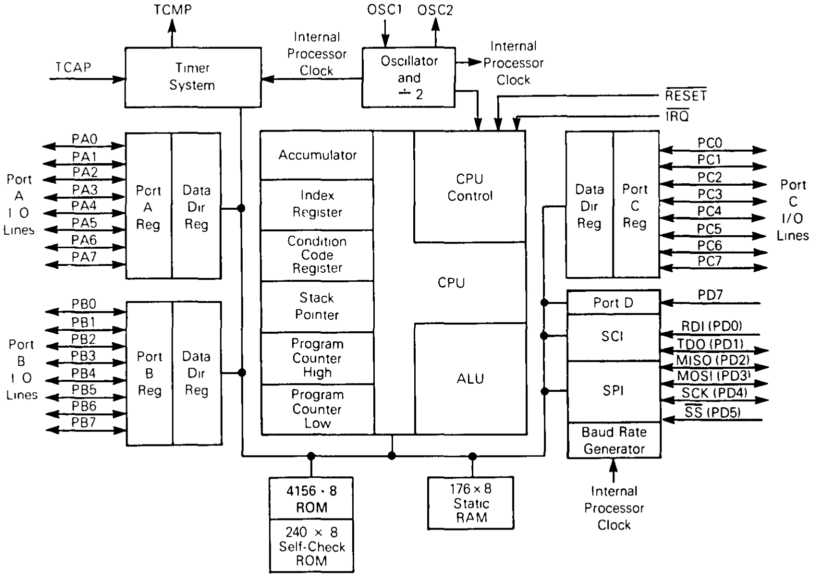
Technical Summary
8-Bit Microcontroller Unit

The MC68HCL05C4 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram for depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory Mapped I/O
- 176 Bytes of On-Chip RAM
- 4156 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8×8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
OSC1
OSC2
Internal Processor Clock
Internal Processor Clock
TCAP
Timer System
Oscillator and ÷ 2
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I O Lines
Port A Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I O Lines
Port B Reg
Data Dir Reg
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
SS (PD5)
Baud Rate Generator
Internal Processor Clock
4156 × 8 ROM
240 × 8 Self-Check ROM
176 × 8 Static RAM

This document contains information on a new product Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{OSC} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance

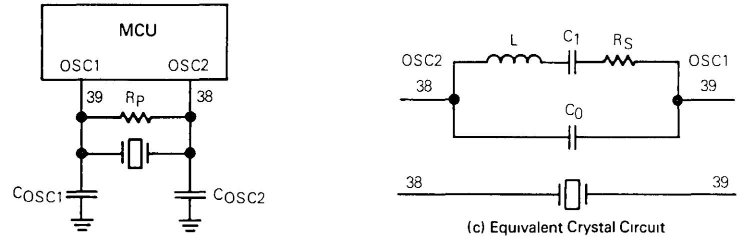
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0 008	0 012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

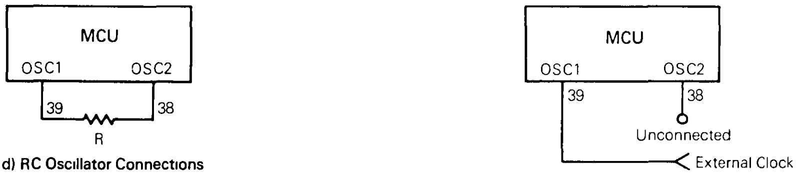
	2-4 MHz	Units
R_S (typical)	10	Ω
C_O	40	pF
C_1	4 3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

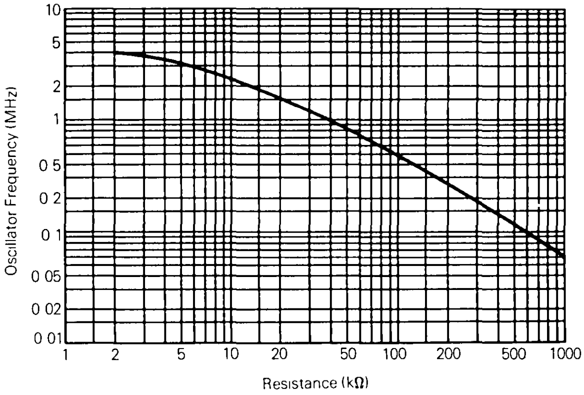


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed-input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

Input/output port progrmming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{\text{W}}$ is an internal signal.

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero.

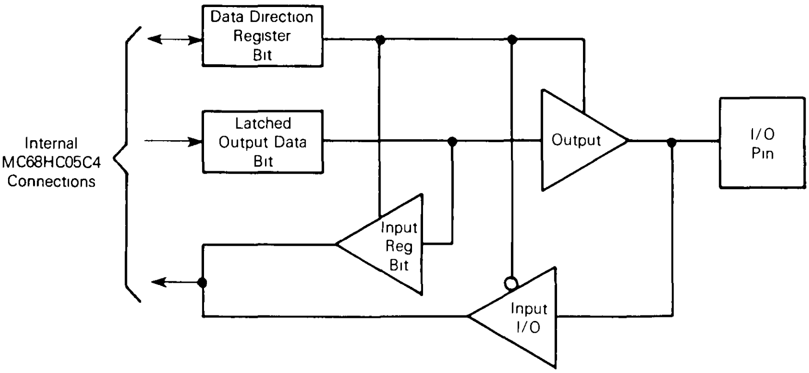


Figure 3. Typical Port I/O Circuit

With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

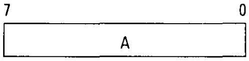
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

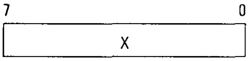
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



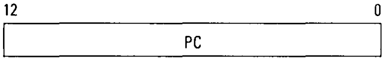
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

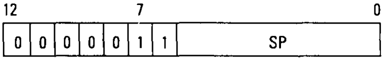
The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack, while an interrupt uses five locations.



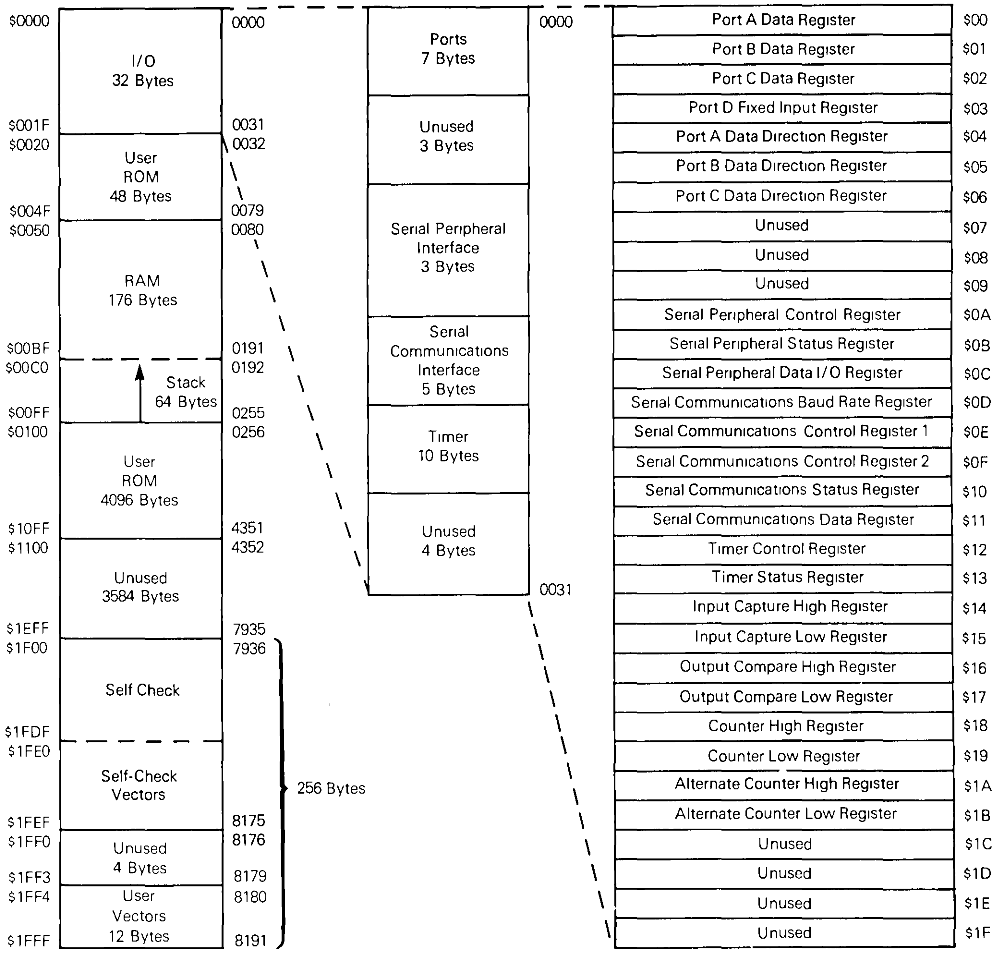

$0000
I/O
32 Bytes
$001F
$0020
User
ROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
User
ROM
4096 Bytes
$10FF
$1100
Unused
3584 Bytes
$1EFF
$1F00
Self Check
$1FDF
$1FE0
Self-Check
Vectors
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
4351
4352
7935
7936
8175
8176
8179
8180
8191
256 Bytes
0000
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Unused
4 Bytes
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Unused $1C
Unused $1D
Unused $1E
Unused $1F

Figure 4. Memory Map

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, and during shifts and rotates.

SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF flag
Interrupts — Tests external, timer, SCI and SPI interrupts
SCI — Transmission test; checks RDRF, TDRE, TC, and FE flags
SPI — Transmission test; checks SPIF, WCOL, and MODF flags

Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

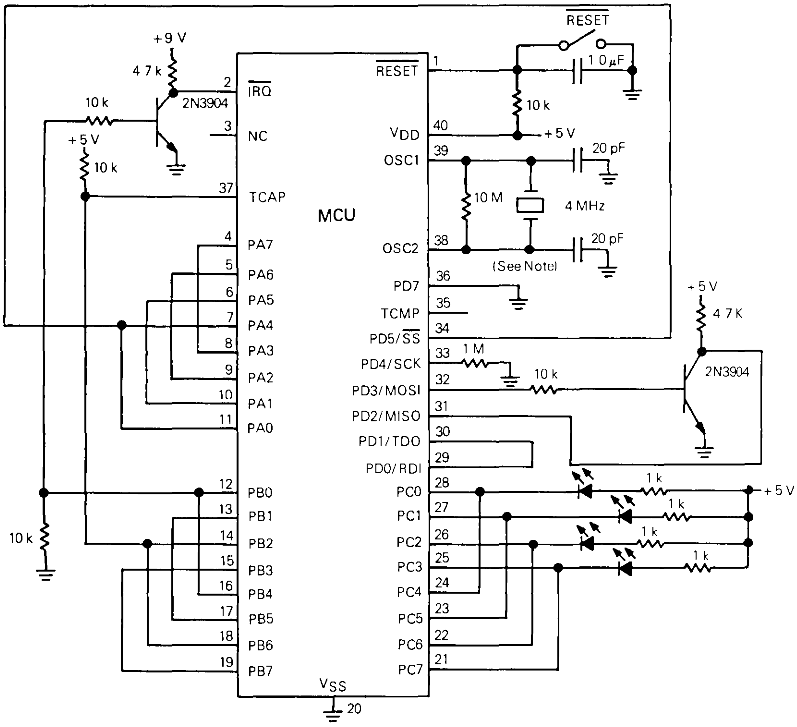
Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or $\overline{\text{IRQ}}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc

0 indicates LED is on; 1 indicates LED is off

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and


+9 V
4.7 k
2
IRQ
2N3904
10 k
3
NC
+5 V
10 k
37
TCAP
MCU
4 PA7
5 PA6
6 PA5
7 PA4
8 PA3
9 PA2
10 PA1
11 PA0
12 PB0
13 PB1
14 PB2
15 PB3
16 PB4
17 PB5
18 PB6
19 PB7
10 k
VSS
20
RESET
1
RESET
1.0 µF
10 k
VDD
40
+5 V
OSC1
39
20 pF
10 M
4 MHz
OSC2
38
20 pF
(See Note)
PD7 36
TCMP 35
PD5/SS 34
PD4/SCK 33
1 M
PD3/MOSI 32
10 k
PD2/MISO 31
PD1/TDO 30
PD0/RDI 29
+5 V
4.7K
2N3904
PC0 28
PC1 27
PC2 26
PC3 25
PC4 24
PC5 23
PC6 22
PC7 21
1 k
1 k
1 k
1 k
+5 V

NOTE The RC Oscillator Option may also be used in this circuit

Figure 5. Self-Check Circuit Schematic Diagram

checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations \$0050 and \$0051 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM checksum subroutine is called at location \$1F93 with RAM location \$0053 equal to \$01 and A=0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations \$0050 through \$0053 are overwritten. Upon return to the user's program, X=0. If the test passed, A=0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 4046 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts ($\overline{\text{IRQ}}$, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

> The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

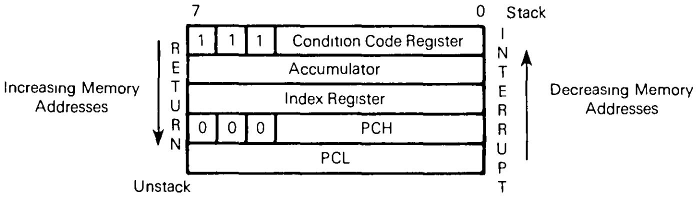
If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of \$1FFA and \$1FFB.



NOTE Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc Pulling from the stack is in the reverse order

Figure 6. Interrupt Stacking Order

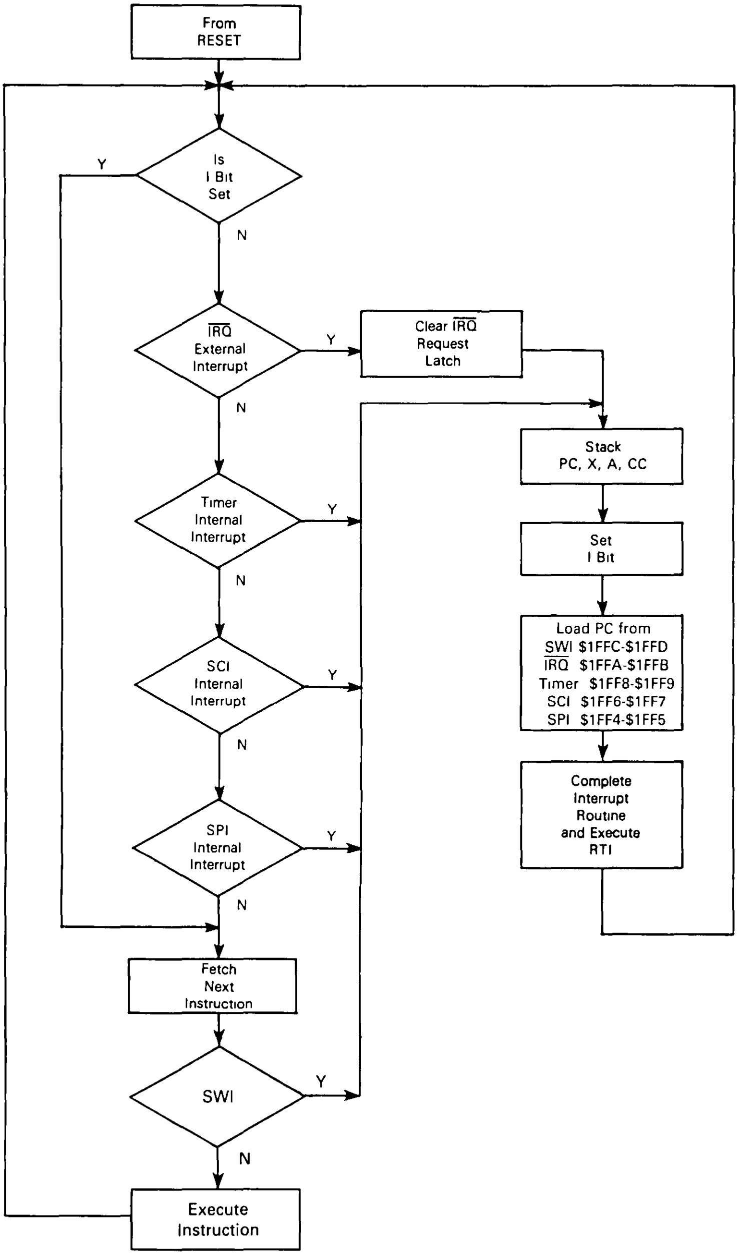
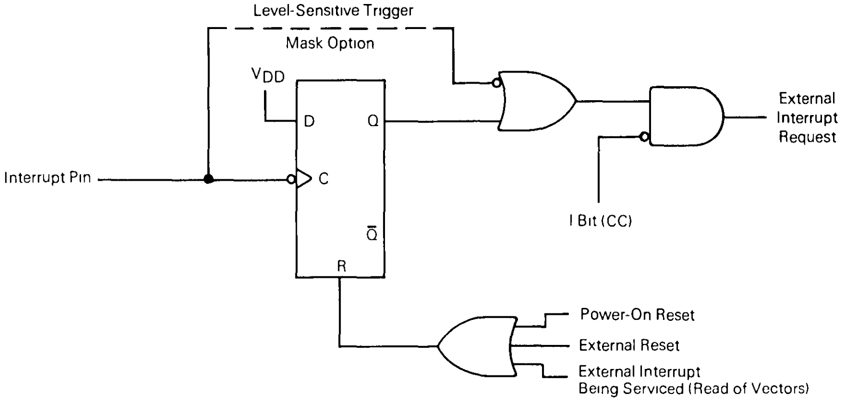

From RESET
Is I Bit Set
Y
N
IRQ External Interrupt
Y
N
Clear IRQ Request Latch
Timer Internal Interrupt
Y
N
SCI Internal Interrupt
Y
N
SPI Internal Interrupt
Y
N
Stack PC, X, A, CC
Set I Bit
Load PC from
SWI $1FFC-$1FFD
IRQ $1FFA-$1FFB
Timer $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI $1FF4-$1FF5
Complete Interrupt Routine and Execute RTI
Fetch Next Instruction
SWI
Y
N
Execute Instruction

Figure 7. Reset and Interrupt Processing Flowchart

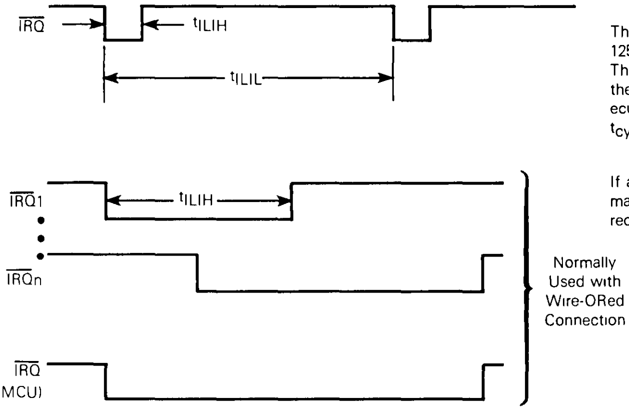
Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.


Level-Sensitive Trigger
Mask Option
V_{DD}
D
Q
C
$\overline{Q}$
R
Interrupt Pin
External Interrupt Request
I Bit (CC)
Power-On Reset
External Reset
External Interrupt Being Serviced (Read of Vectors)

(a) Interrupt Function Diagram


$\overline{IRQ}$
t_{ILIH}
t_{ILIL}
$\overline{IRQ}1$
t_{ILIH}
$\overline{IRQ}n$
$\overline{IRQ}$ (MCU)
Normally Used with Wire-ORed Connection

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 8. External Interrupt

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

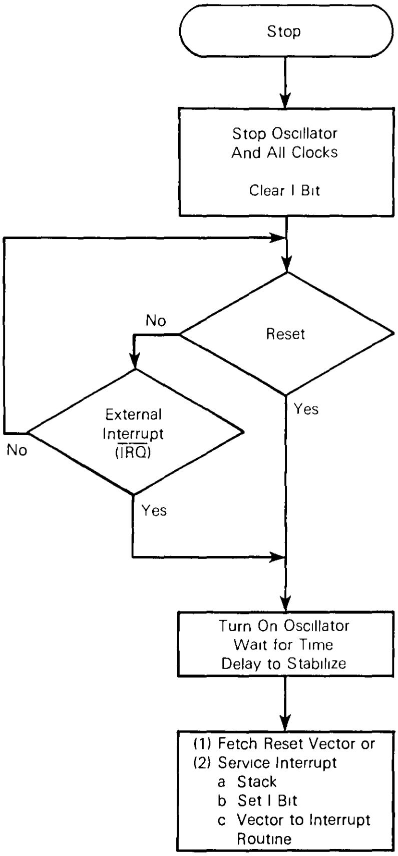

Stop
Stop Oscillator And All Clocks
Clear I Bit
Reset
No
Yes
External Interrupt ($\overline{IRQ}$)
No
Yes
Turn On Oscillator Wait for Time Delay to Stabilize
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt Routine

Figure 9. STOP Function Flowchart

SCI During STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{IRQ}$ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI During STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the $\overline{IRQ}$ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active; refer to Figure 10. An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{RESET}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler which divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations \$18–\$19 (counter register) or \$1A–\$1B, (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter (\$19, \$1B) receives the count value at the time of the read. If a read of the free running counter or counter alternate register first addresses the most significant byte (MSB) (\$18, \$1A), the LSB (\$19, \$1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB (\$19 or \$1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to \$FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to \$FFFC, and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from \$FFFF to \$0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations \$16 (MSB) and \$17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

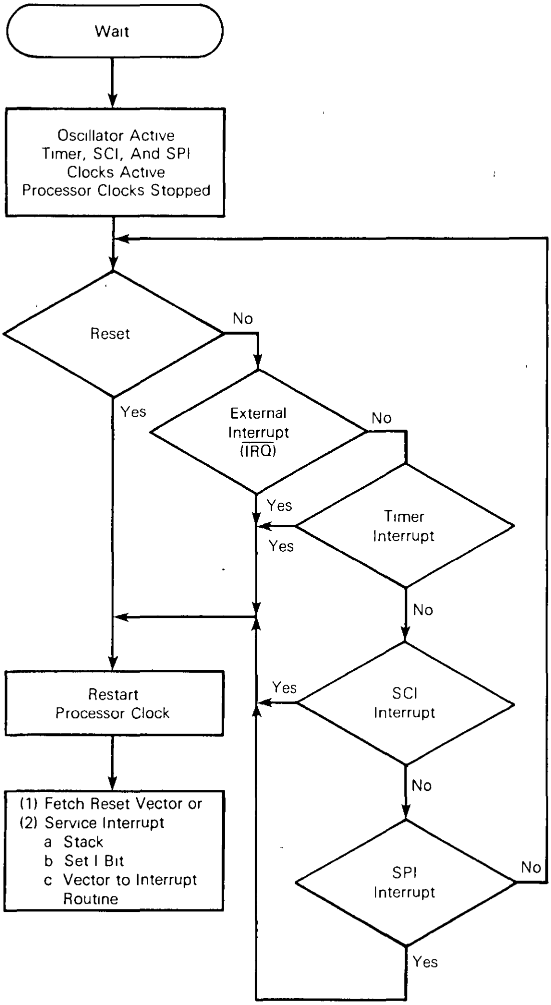

Wait
Oscillator Active
Timer, SCI, And SPI
Clocks Active
Processor Clocks Stopped
Reset
No
Yes
External
Interrupt
($\overline{IRQ}$)
No
Yes
Timer
Interrupt
Yes
No
Restart
Processor Clock
Yes
SCI
Interrupt
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt
Routine
SPI
Interrupt
No
Yes

Figure 10. WAIT Function Flowchart

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free running counter after the corresponding

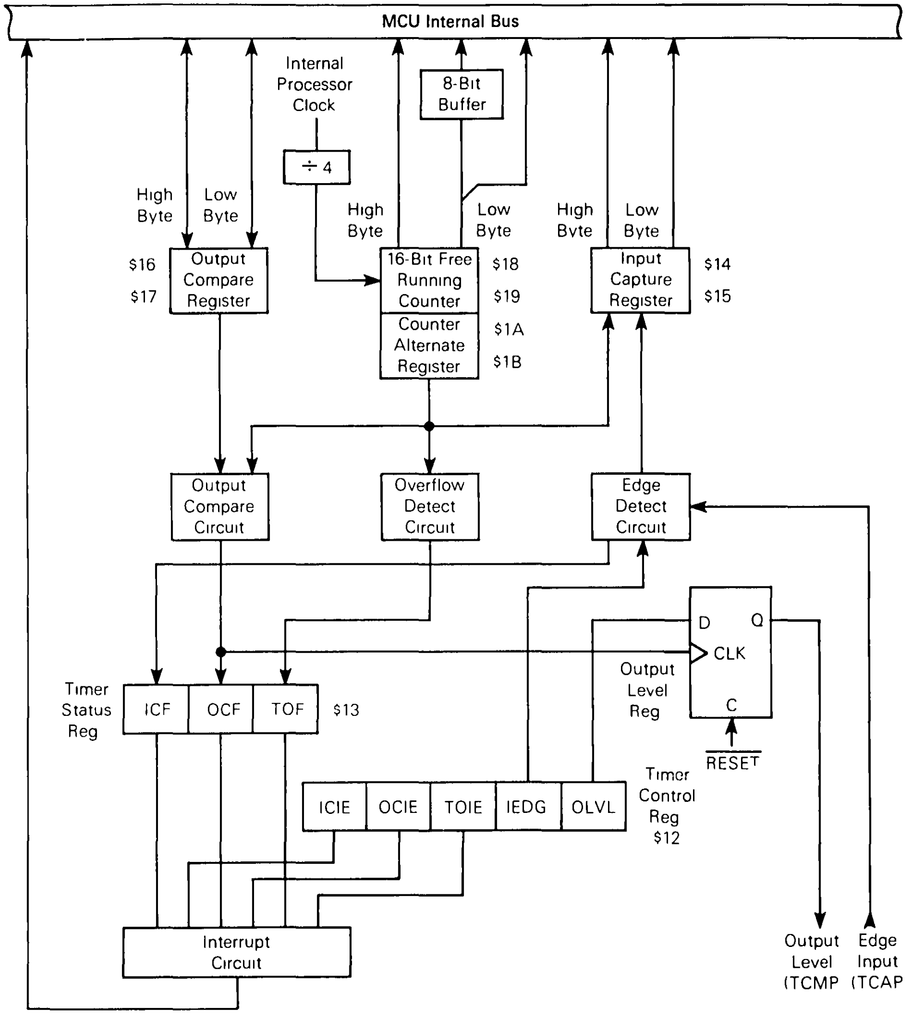

MCU Internal Bus
Internal Processor Clock
÷ 4
8-Bit Buffer
High Byte
Low Byte
High Byte
Low Byte
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
Counter Alternate Register
$18
$19
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
Timer Control Reg $12
Interrupt Circuit
Output Level (TCMP
Edge Input (TCAP

Figure 11. Timer Block Diagram

input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value which corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET.	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
OCIE — Output Compare Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
TOIE — Timer Overflow Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
IEDG — Input Edge
  Value of input edge determines which level transition on TCAP pin will trigger free running counter transfer to the input capture register
  1 = Positive edge
  0 = Negative edge
  Reset does not affect te IEDG bit (U = unaffected).
OLVL — Output Level
  Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
  1 = High output
  0 = Low output
Bits 2, 3, and 4 — Not used
  Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
  1 = Flag set when selected polarity edge is sensed by input capture edge detector
  0 = Flag cleared when TSR and input capture low register ($15) are accessed
OCF — Output Compare Flag
  1 = Flag set when output compare register contents match the free running counter contents
  0 = Flag cleared when TSR and output compare low register ($17) are accessed
TOF — Timer Overflow Flag
  1 = Flag set when free running counter transition from $FFFF to $0000 occurs
  0 = Flag cleared when TSR and counter low register ($19) are accessed
Bits 0–4 — Not used
  Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If reset is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If reset is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software selectable word length (eight or nine bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data which is presented between the internal data bus and the output pin (TDO), and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 12.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is reenabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate which is 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 13); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data which is presented from the transmit data register (TDR), via the SCI, to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to 1/16 that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 13. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits which individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable and send break code bits. The baud rate register bits allow the user to select different baud rates which are used as the rate control for the transmitter and receiver.

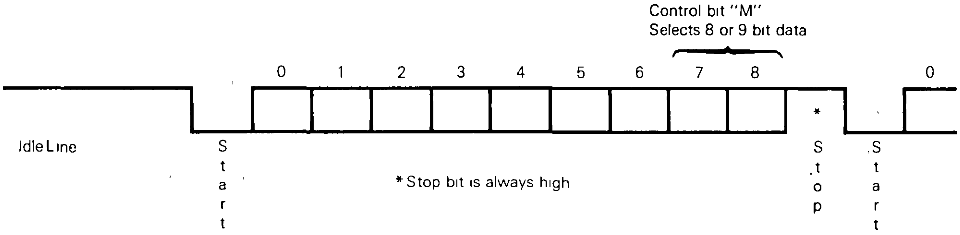
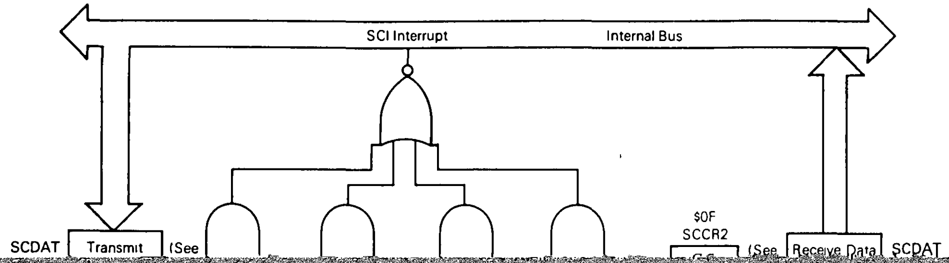


Figure 12. Data Format


SCI Interrupt
Internal Bus
SCDAT
Transmit
(See
$0F
SCCR2
(See
Receive Data
SCDAT

receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

7	6	5	4	3	2	1	0
SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET							
U	U	U	U	U	U	U	U

As shown in Figure 13, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $OE

The SCCR1 provides control bits which determine word length and select the wake-up method.

7	6	5	4	3	2	1	0
R8	T8	—	M	WAKE	—	—	—
RESET							
U	U	—	U	U	—	—	—

R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most-significant bit)
0 = Idle line condition

Bits 0–2, and 5 — Not used
Can read either one or zero

The address bit is dependent on both the wake bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $OF

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:							
0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt enabled
0 = TDRE interrupt disabled

TCIE — Transmit Complete Interrupt Enable
1 = SCI interrupt enabled
0 = TC interrupt disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt enabled
0 = RDRF and OR interrupts disabled

ILIE — Idle Line Interrupt Enable
1 = SCI interrupt enabled
0 = Idle interrupt disabled

TE — Transmit Enable
1 = Transmit shift register output applied to the TDO line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
0 = Transmitter disabled after last byte loaded in the SCDAT and TDRE set. After last byte is transmitted, TDO line becomes a high-impedance line.

RE — Receive Enable
1 = Receiver shift register input applied to the RDI line
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits inhibited

RWU — Receiver Wake Up
1 = Places receiver in sleep mode and enables wake-up function
0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)

SBK — Send Break
1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.

0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET.	1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
  1 = TDR contents transferred to the transmit data shift register
  0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1) followed by a write to the TDR.

TC — Transmit Complete
  1 = Indicates end of data frame, preamble, or break condition has occurred
  0 = TC bit cleared by reading the SCSR (with TC = 1) followed by a write to the TDR

RDRF — Receive Data Register (RDR) Full
  1 = Receiver data shift register contents transferred to the RDR
  0 = Receiver data shift register transfer did not occur. RDRF cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR

IDLE — Idle Line Detect
  1 = Indicates receiver has detected an idle line
  0 = IDLE cleared by reading the SCSR (with IDLE = 1) followed by a read of the RDR. Once cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error
  1 = Indicates receive data shift register data sent to a full RDR (RDRF = 1). Data causing the overrun is lost and RDR data is not disturbed.
  0 = OR cleared by reading the SCSR (with OR = 1) followed by a read of the RDR

NF — Noise Flag
  1 = Indicates noise present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
  0 = NF cleared by reading the SCSR (with NF = 1) followed by a read of the RDR

FE — Framing Error
  1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.
  0 = NF cleared by reading the SCSR (with FE = 1) followed by a read of the RDR

Bit 0 — Not used
  Can read either one or zero

Baud Rate Register $0D

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600 Hz baud rate is required with a 2.4576 MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 14), or MCUs that can be either masters or slaves.

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	**0**		**4.194304**	**4.0**	**2.4576**	**2.0**	**1.8432**
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receive clock is 16 times higher in frequency than the actual baud rate.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programamble master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most-significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}=1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 15, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

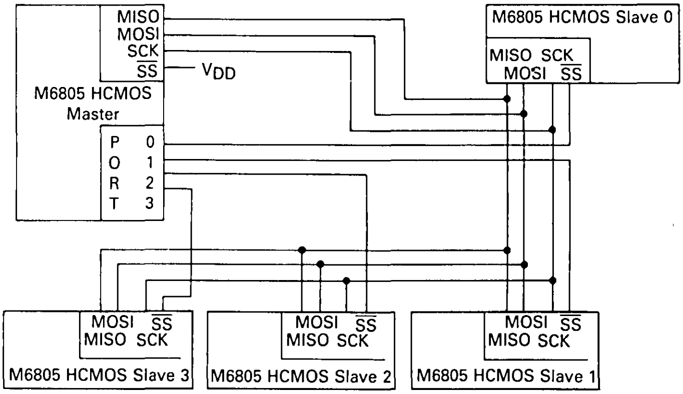


Figure 14. Master-Slave System Configuration

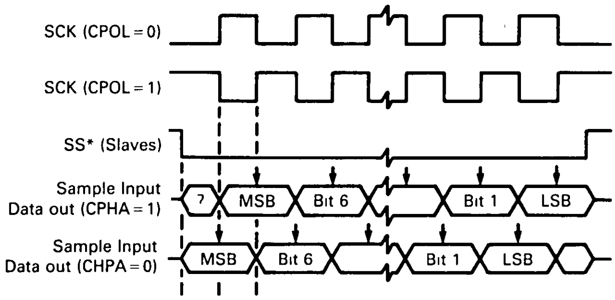

SCK (CPOL = 0)
SCK (CPOL = 1)
SS* (Slaves)
Sample Input
Data out (CPHA = 1)
7
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA = 0)
MSB
Bit 6
Bit 1
LSB

Figure 15. Data Clock Timing Diagram

Slave Select

The slave select (SS) input line selects a slave device. The SS line must be low prior to data transactions and must stay low for the duration of the transaction. The SS line on the master must be tied high; if the SS line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA = 0, the shift clock is the OR of SS with SCK. In this clock phase mode, SS must go high between successive characters in an SPI message. When CPHA = 1, SS must go high between successive characters in an SPI message. When CPHA = 1, SS may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU SS line could be tied to V_{SS} as long as CPHA = 1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 16. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the SS pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data

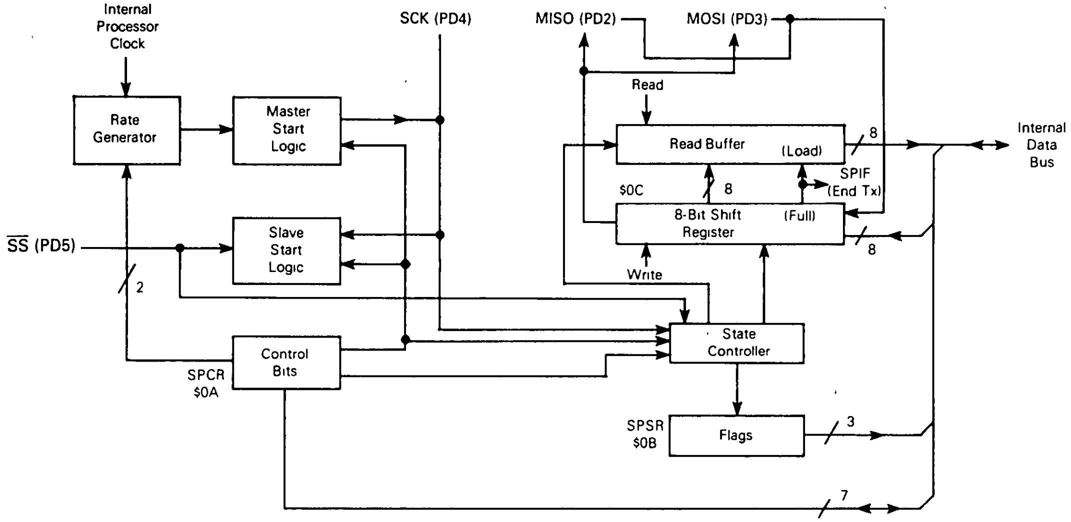

Internal Processor Clock
Rate Generator
Master Start Logic
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Read
Read Buffer
(Load)
8
Internal Data Bus
SPIF (End Tx)
$0C
8
8-Bit Shift Register
(Full)
8
SS (PD5)
Slave Start Logic
2
Write
State Controller
Control Bits
SPCR $0A
SPSR $0B
Flags
3
7

Figure 16. SPI Block Diagram

from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 17 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register (SPCR) $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

7	6	5	4	3	2	1	0
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET							
0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase
Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits
Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. If in the slave mode the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 — Not used
Can read either one or zero

SPI Clock Rate Selection

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR) $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—
RESET							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device (If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing accomplished by reading SPSR (with SPIF = 1) followed by SPDR access

WCOL — Write Collision
1 = Indicates an attempt made to write to SPDR while data transfer is in process
0 = Clearing accomplished by reading SPSR (with WCOL = 1) followed by SPDR access

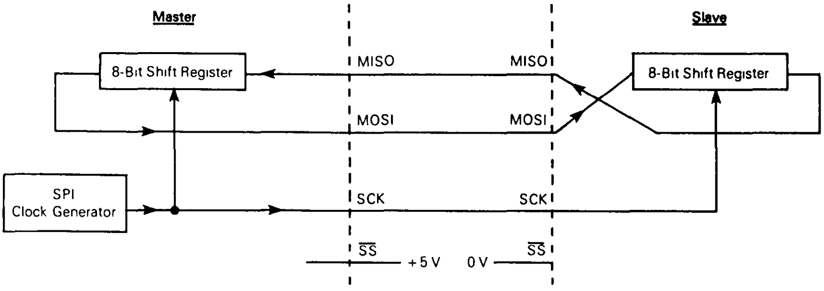


Figure 17. SPI Master-Slave Interconnections

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict
0 = Clearing accomplished by reading SPSR (with MODF = 1) followed by a write to the SPCR

Bits 0–3, and 5 — Not used
Can read either zero or one

Serial Peripheral Data I/O Register (SPDR) $0C

The SPDR is a read/write register used to receive and transmit SPI data.

	7	6	5	4	3	2	1	0
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET:	U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register			
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n = 0 . . . 7)
Set Bit n	BSET n (n = 0 . . 7)
Clear Bit n	BCLR n (n = 0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

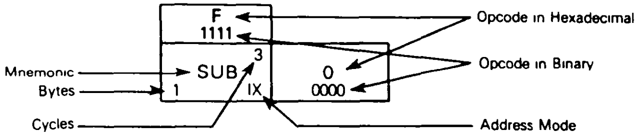
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. The using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to $+7.0$	V
Input Voltage	V_{in}	V_{SS} -0.3 to V_{DD} $+0.3$	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	$V_{SS} - 0.3$ to $2 \times V_{DD} + 0.3$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HCL05C4 (Standard)	T_A	T_L to T_H 0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to $+150$	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		60	
Plastic Leaded Chip Carrier (PLCC)		70	

$V_{DD} = 4.5$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0, PD5,PD7	1.9 kΩ	2.26 kΩ	200 pF

$V_{DD} = 3.0$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ	6.32 kΩ	50 pF
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF

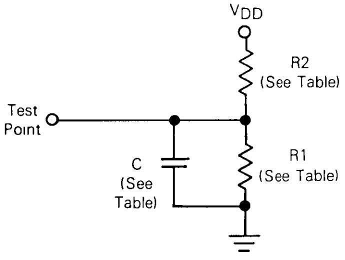


Figure 18. Equivalent Test Load

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.5 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD}-0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD}-0.8$ $V_{DD}-0.8$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 23) Wait (see Figures 22 and 23) Stop (see Figure 23) 25°C 0° to 70°C	I_{DD}	 — — — —	 — — — —	 5.0 2.75 15 25	 mA mA μA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	± 1.0	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	± 1	μA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 2.1 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.4 Vdc–3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD}-0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 0.4 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD}-0.3$ $V_{DD}-0.3$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (3.6 Vdc at 1.0 MHz) Run (See Figures 22 and 24) Wait (See Figures 22 and 24) Stop (See Figure 24) 25°C 0°C to 70°C	I_{DD}	 — — — —	 — — — —	 1.75 900 5.0 10	 mA µA µA µA
Supply Current (2.4 Vdc at 500 kHz) Run (See Figures 22 and 25) Wait (See Figures 22 and 25) Stop (See Figure 25) 25°C 0° to 70°C	I_{DD}	 — — — —	 — — — —	 750 400 2.0 5.0	 µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±1.0	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source all inputs 0.2 V from rail, no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

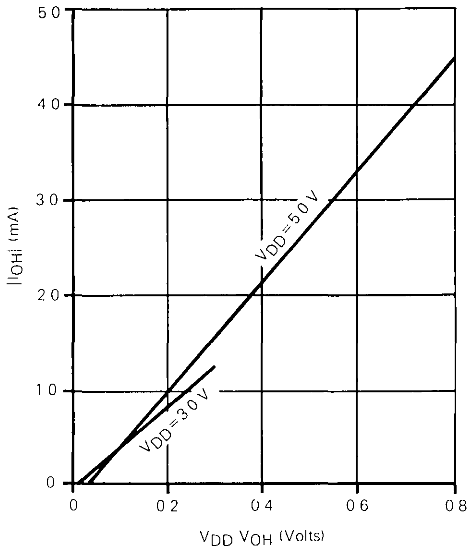


Figure 19. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

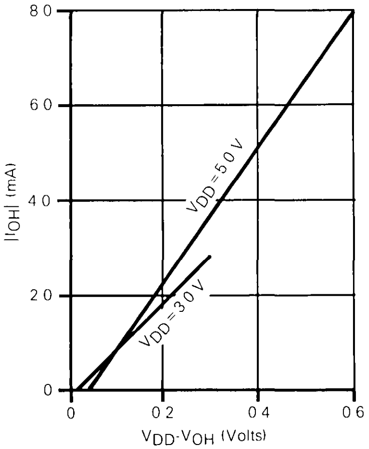


Figure 20. Typical V_{OH} vs I_{OH} for PD1-PD4

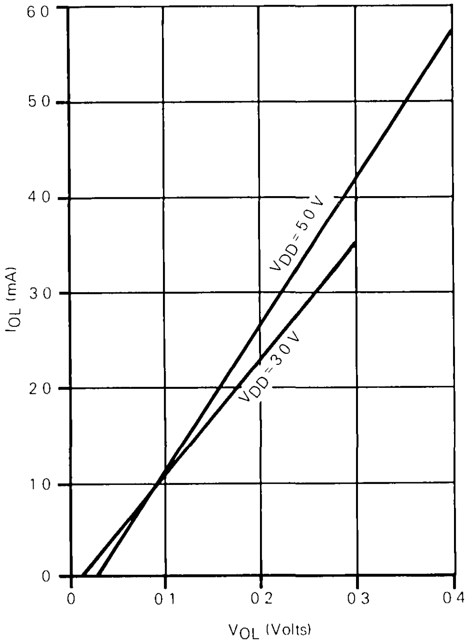


Figure 21. Typical V_{OL} vs I_{OL} for all Ports

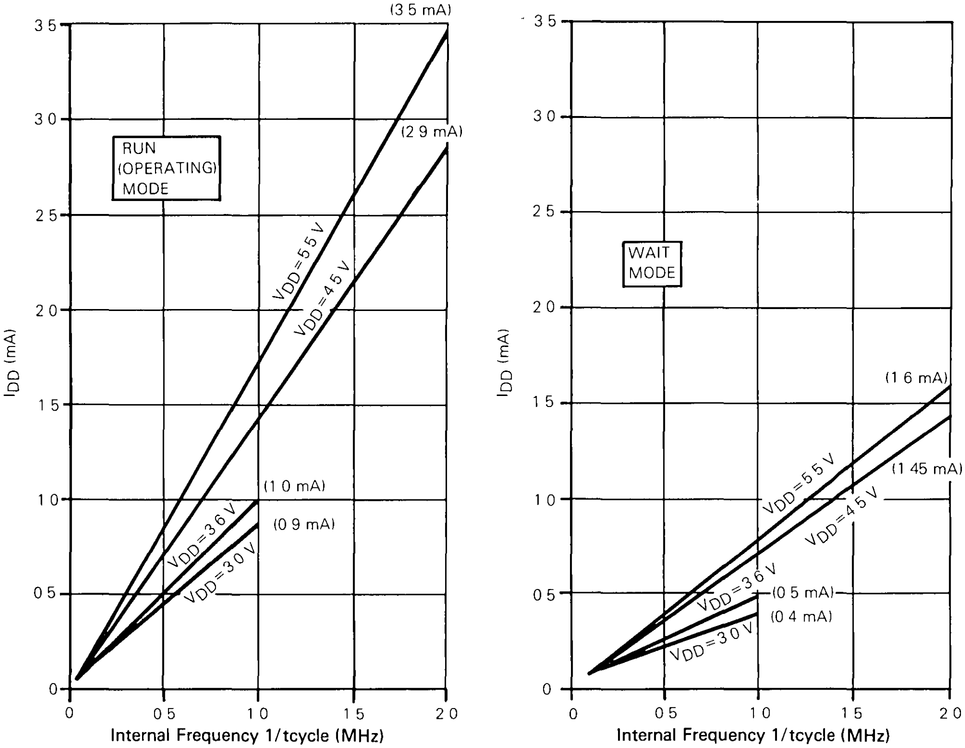


Figure 22. Typical Current vs Internal Frequency for Run and Wait Modes

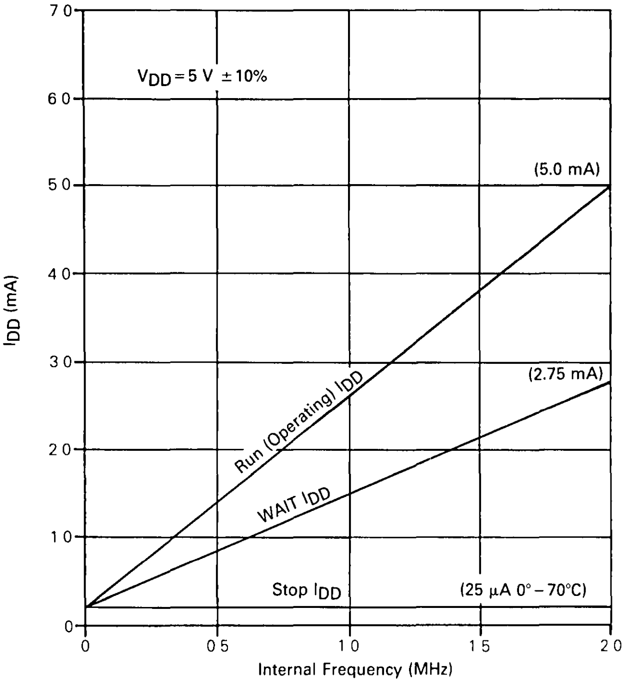


Figure 23. Maximum I_{DD} vs Frequency for $V_{DD}=5.5$ Vdc

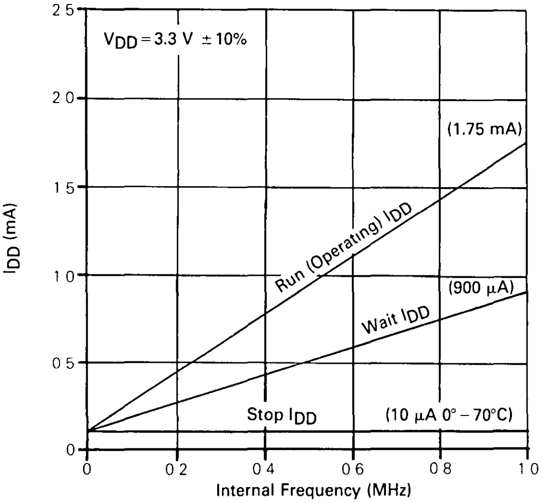


Figure 24. Maximum I_{DD} vs Frequency for $V_{DD}=3.6$ Vdc

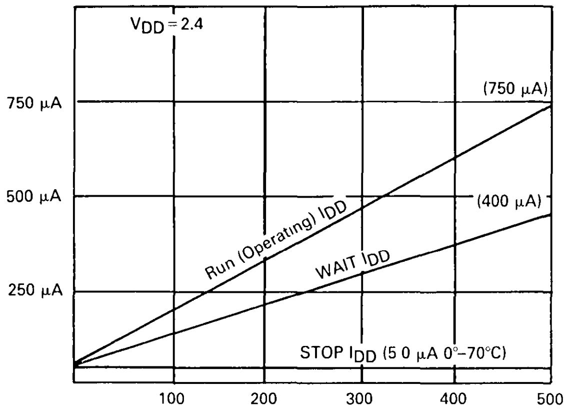


Figure 25. Maximum I_{DD} vs Frequency for $V_{DD} = 2.4$ Vdc

CONTROL TIMING

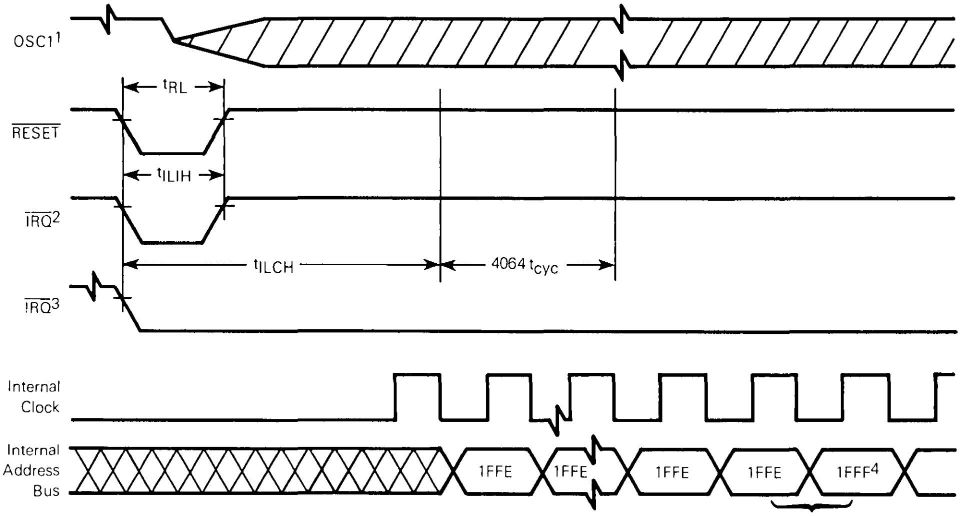
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} - 2$)		—	2.1	
External Clock ($f_{osc} - 2$)		dc	2.1	
Cycle Time (see Figure 29)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 29)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 29)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 27)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 27)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3 $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option
4 $\overline{\text{RESET}}$ vector address shown for timing example

Figure 26. Stop Recovery Timing Diagram

CONTROL TIMING

($V_{DD}=2.4$ Vdc–3.6 Vdc, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 2.0 2.0	MHz
Internal Operating Frequency Crystal ($f_{osc}-2$) External Clock ($f_{osc}-2$)	f_{op}	 — dc	 1.0 1.0	MHz
Cycle Time (see Figure 29)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 29)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 29)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 27) Input Capture Pulse Period (see Figure 27)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

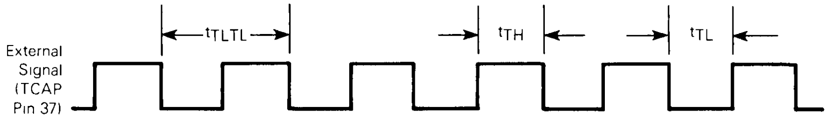


Figure 27. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 240	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

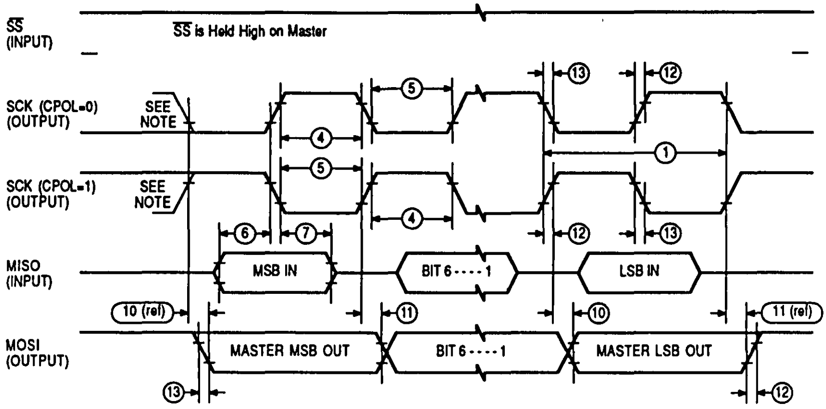
SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 2.4 Vdc–3.6 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	µs
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	720	—	µs
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	720	—	µs
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	250	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	500	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

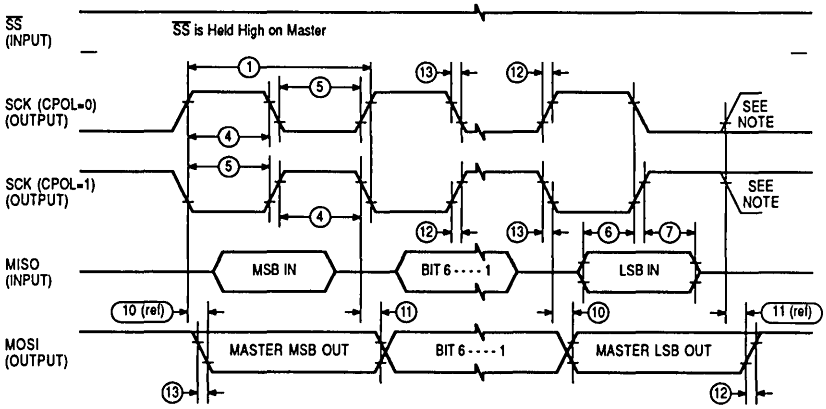
*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

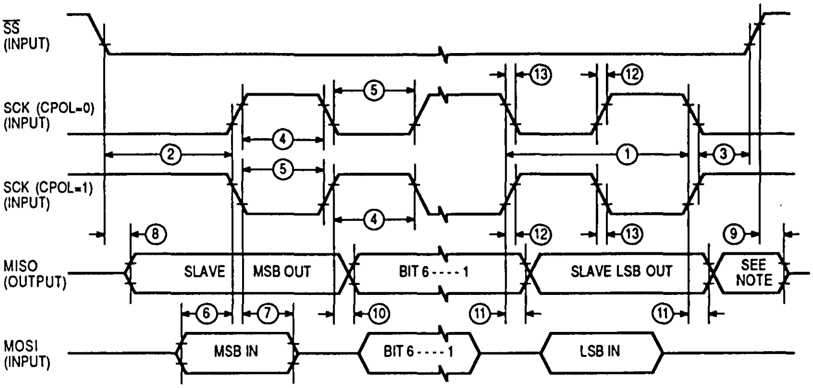
a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

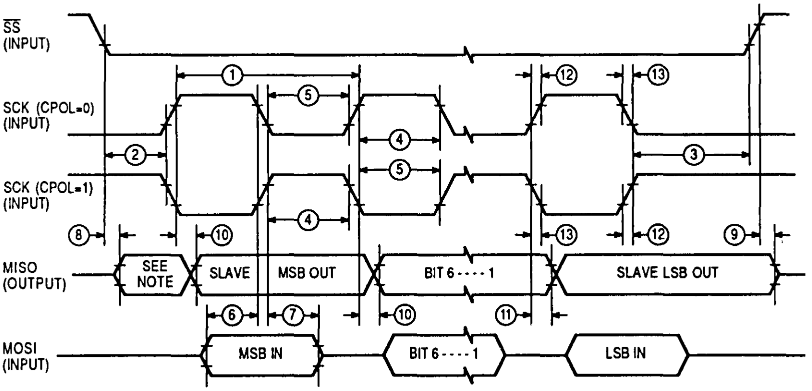
b) SPI MASTER TIMING (CPHA=1)

Figure 28. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE Not defined but normally MSB of character just received.

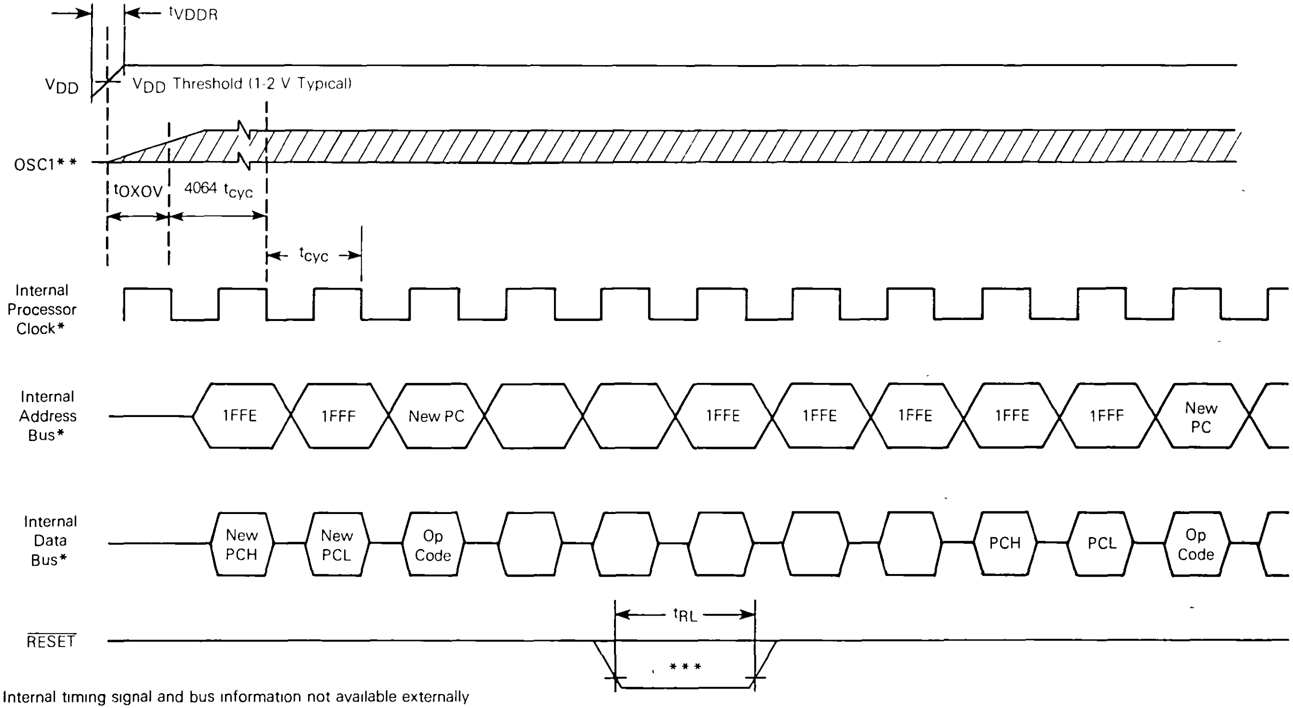
c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE. Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA=1)

Figure 28. SPI Timing Diagrams (Sheet 2 of 2)


tVDDR
VDD
VDD Threshold (1-2 V Typical)
OSC1**
tOXOV
4064 tcyc
tcyc
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
tRL
RESET
***

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 29. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS, disk file
MS-DOS/PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM MC68HC805C4

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS®/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

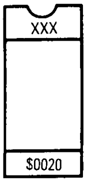
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2764, 68764, or 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

For an MC68HCL805C4 MCU, start the page zero, user ROM at EEPROM address $0020 through $004F. Start the user ROM at EEPROM address $0100 through $10FF with vectors from $1FF4 to $1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

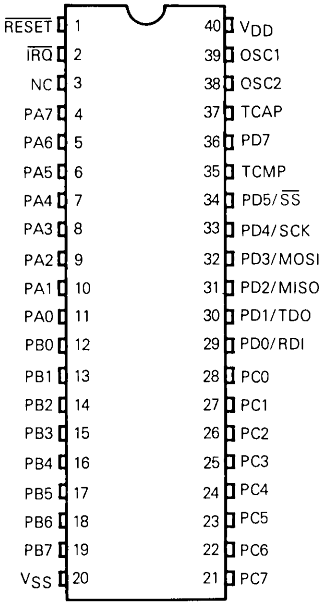
ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HCL05C4 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to 70°C	MC68HCL05C4P
PLCC (FN Suffix)	0°C to 70°C	MC68HCL05C4FN

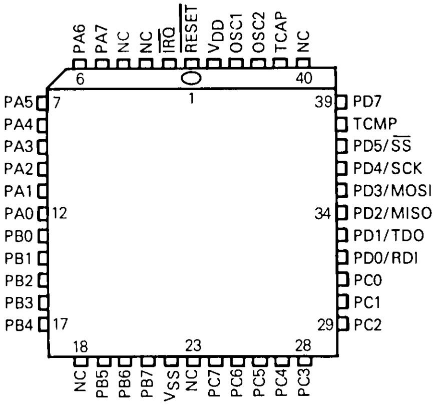
PACKAGE DIMENSIONS

40-PIN DUAL-IN-LINE PACKAGE



RESET 1
IRQ 2
NC 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
V_{SS} 20
40 V_{DD}
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE



PA6
PA7
NC
NC
IRQ
RESET
V_{DD}
OSC1
OSC2
TCAP
NC
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
V_{SS}
NC
PC7
PC6
PC5
PC4
PC3

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HCL05C8

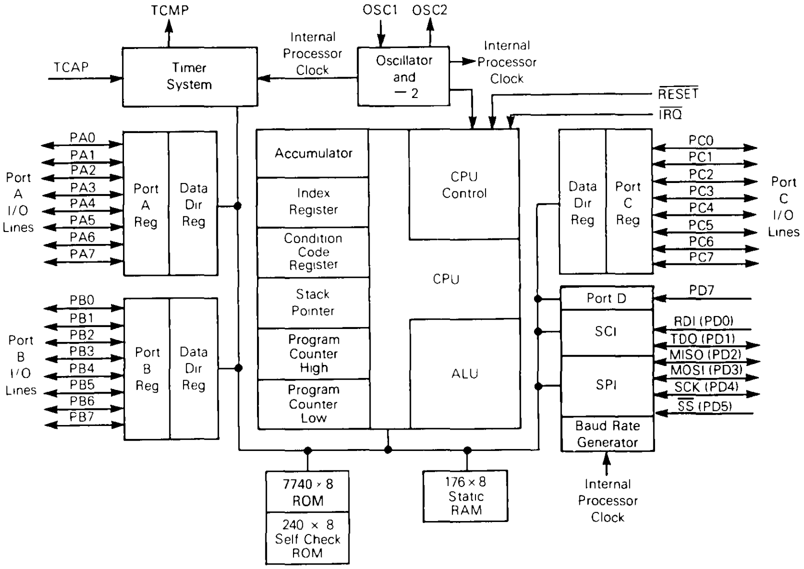
Technical Summary
8-Bit Microcontroller Unit

The MC68HCL05C8 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 7740 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8×8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
OSC1
OSC2
Internal Processor Clock
Internal Processor Clock
TCAP
Timer System
Oscillator and ÷ 2
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I/O Lines
Port A Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I/O Lines
Port B Reg
Data Dir Reg
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
SS (PD5)
Baud Rate Generator
Internal Processor Clock
7740 × 8 ROM
240 × 8 Self Check ROM
176 × 8 Static RAM

This document contains information on a new product Specifications and information herein are subject to change without notice

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

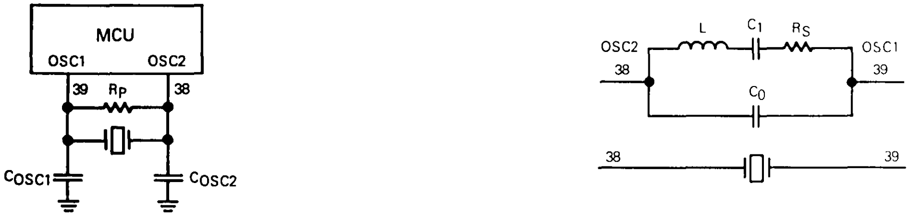
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

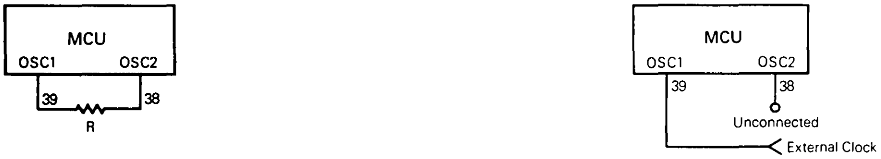
	2-4 MHz	Units
R_S (typical)	10	Ω
C_O	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

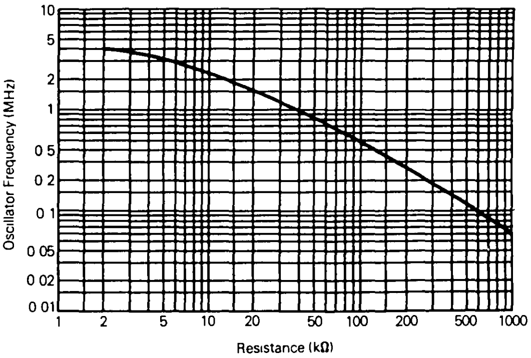


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{\text{W}}$ is an internal signal.

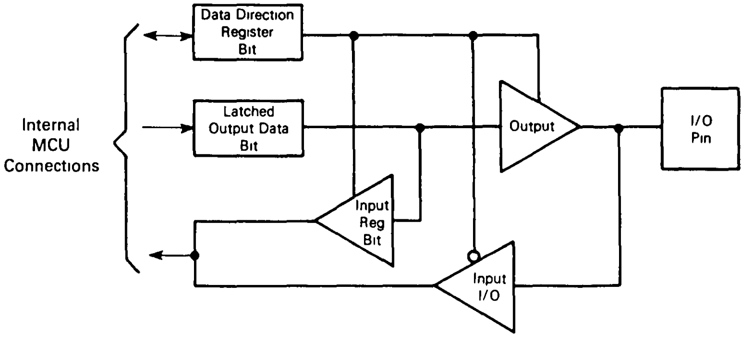


Figure 3. Typical Port I/O Circuit

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

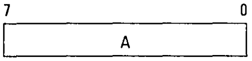
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

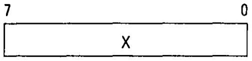
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



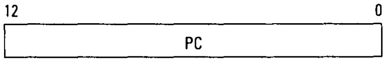
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0.

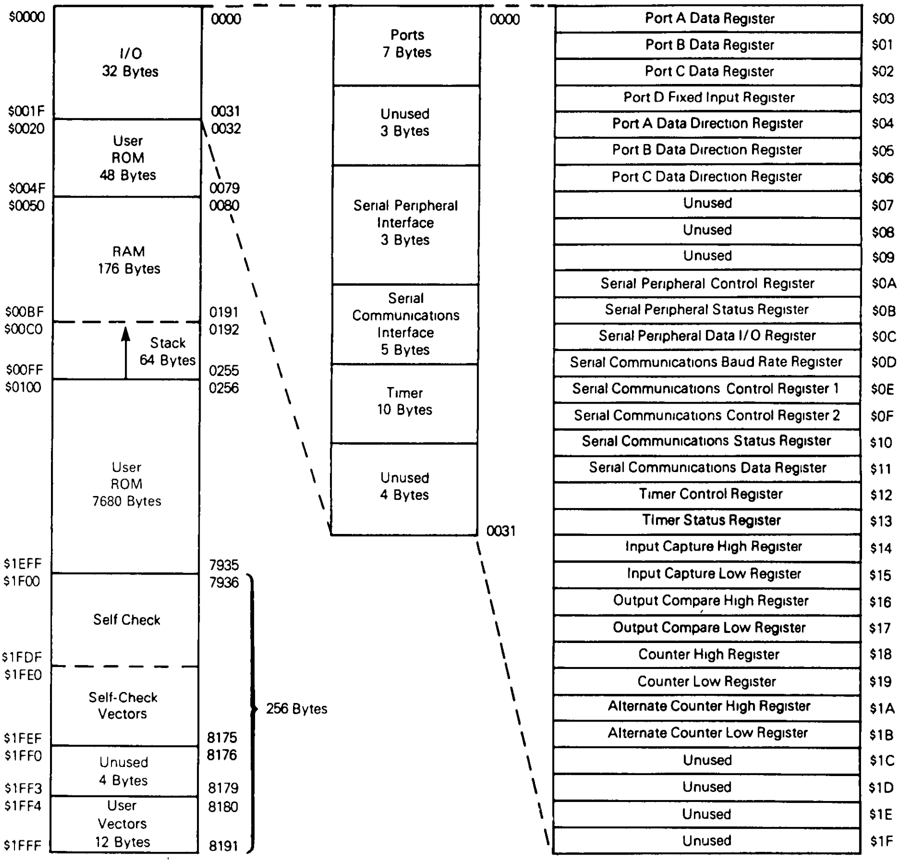
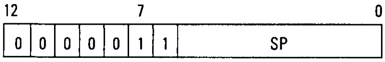

$0000
I/O
32 Bytes
$001F
$0020
User
ROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
User
ROM
7680 Bytes
$1EFF
$1F00
Self Check
$1FDF
$1FE0
Self-Check
Vectors
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
7935
7936
8175
8176
8179
8180
8191
256 Bytes
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Unused
4 Bytes
0000
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Unused $1C
Unused $1D
Unused $1E
Unused $1F

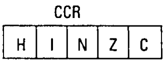
Figure 4. Memory Map

Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.


12
7
0
0
0
0
0
0
1
1
SP

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF flag
Interrupts — Tests external, timer, SCI and SPI interrupts
SCI — Transmission test; checks RDRF, TDRE, TC, and FE flags
SPI — Transmission test; checks SPIF, WCOL, and MODF flags

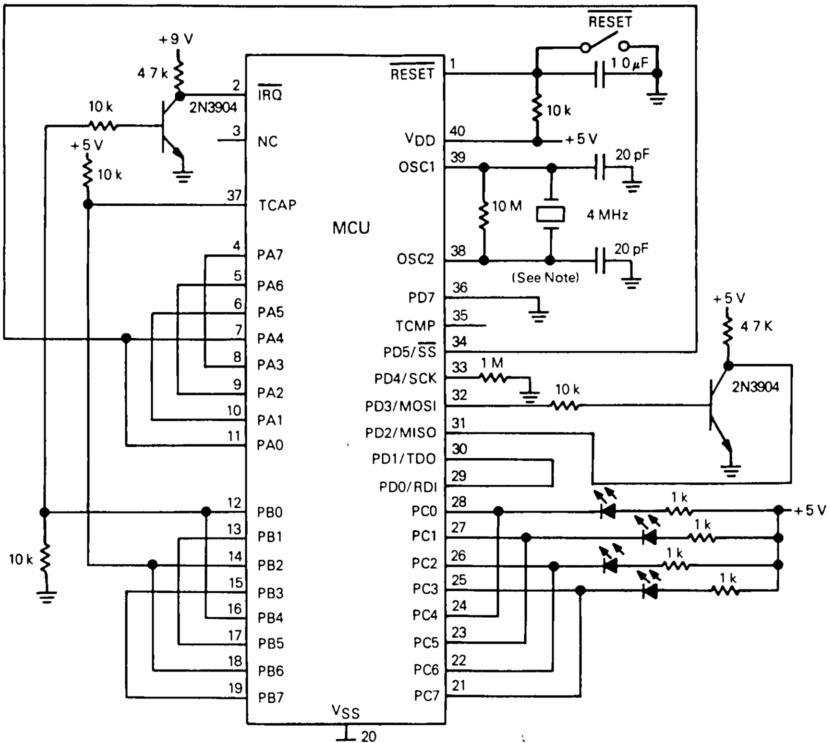
Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X = 40. If the test passed, A = 0.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM


+9 V
4 7 k
10 k
2N3904
+5 V
10 k
10 k
IRQ
NC
TCAP
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
MCU
VSS
20
RESET
VDD
OSC1
OSC2
PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
RESET
1 0 μF
10 k
+5 V
20 pF
10 M
4 MHz
20 pF
(See Note)
1 M
10 k
+5 V
4 7 K
2N3904
1 k
1 k
1 k
1 k
+5 V

NOTE The RC Oscillator Option may also be used in this circuit

Figure 5. Self-Check Circuit Schematic Diagram

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or IRQ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on, 1 indicates LED is off.

checksum subroutine is called at location \$1F93 with RAM location \$0053 equal to \$01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations \$0050 through \$0053 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input (RESET). The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the RESET pin is low at the end of 4046 t_{cyc}, the MCU will remain in the reset condition until RESET goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts (IRQ, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

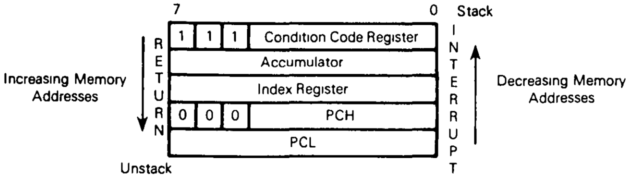
If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer



NOTE Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order

Figure 6. Interrupt Stacking Order

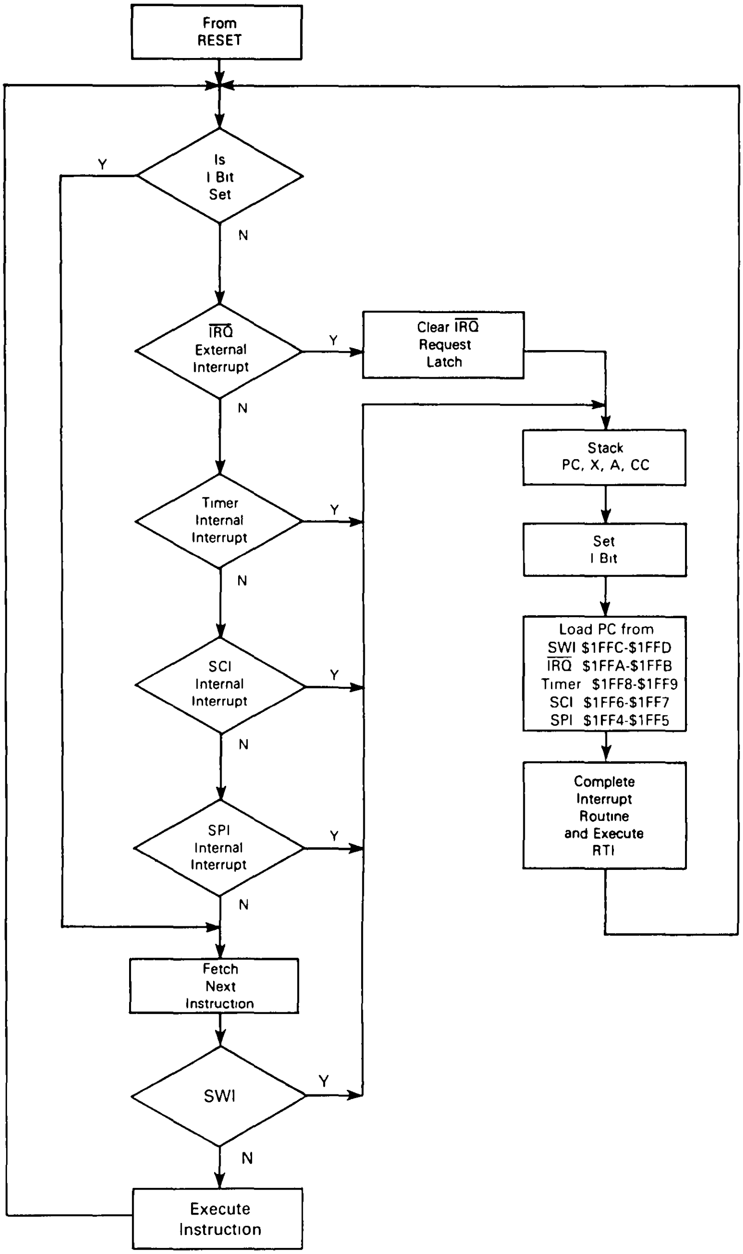

From RESET
Is I Bit Set
Y
N
IRQ External Interrupt
Y
Clear IRQ Request Latch
N
Stack PC, X, A, CC
Timer Internal Interrupt
Y
N
Set I Bit
SCI Internal Interrupt
Y
N
Load PC from
SWI $1FFC-$1FFD
IRQ $1FFA-$1FFB
Timer $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI $1FF4-$1FF5
SPI Internal Interrupt
Y
N
Complete Interrupt Routine and Execute RTI
Fetch Next Instruction
SWI
Y
N
Execute Instruction

Figure 7. Reset and Interrupt Processing Flowchart

interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of \$1FFA and \$1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

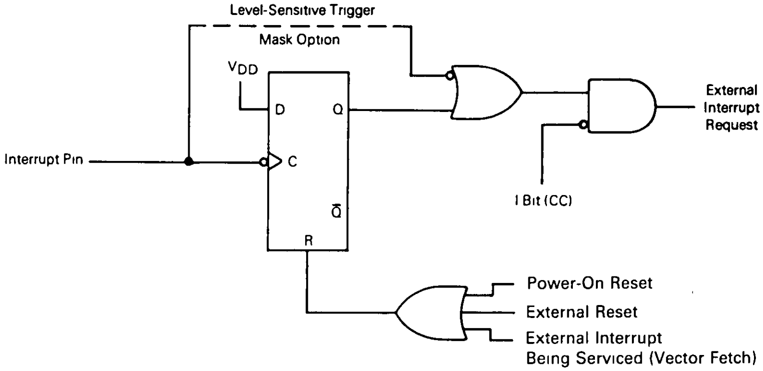
The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT (SWI)

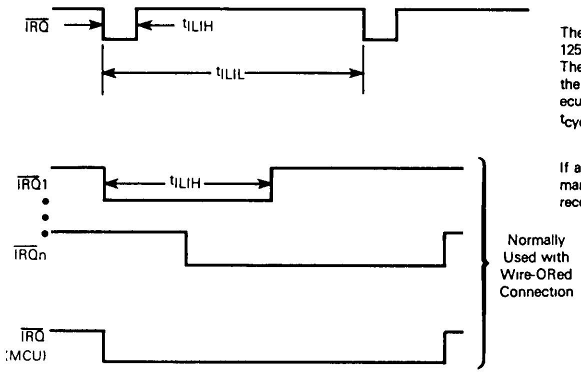
The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations \$1FFC and \$1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set.



(a) Interrupt Internal Function Diagram



Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is either 125 ns ($V_{DD}=5$ V) or 250 ns ($V_{DD}=3$ V) The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 8. External Interrupt

Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI during STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{IRQ}$ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the $\overline{IRQ}$ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the $\overline{IRQ}$ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

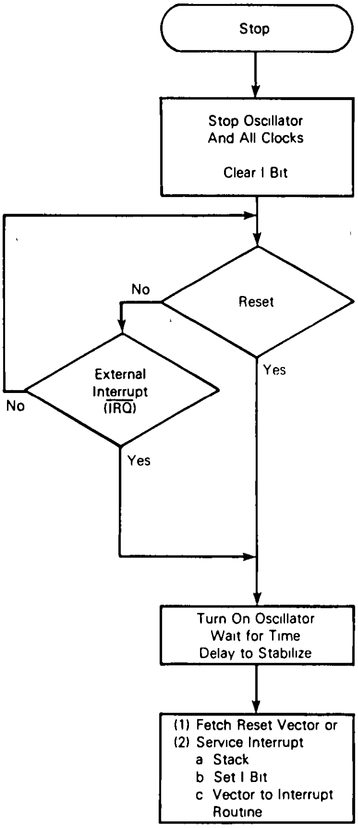

Stop
Stop Oscillator And All Clocks
Clear I Bit
Reset
No
Yes
External Interrupt ($\overline{IRQ}$)
No
Yes
Turn On Oscillator Wait for Time Delay to Stabilize
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt Routine

Figure 9. STOP Function Flowchart

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active (refer to Figure 10). An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{RESET}$ during data retention mode.

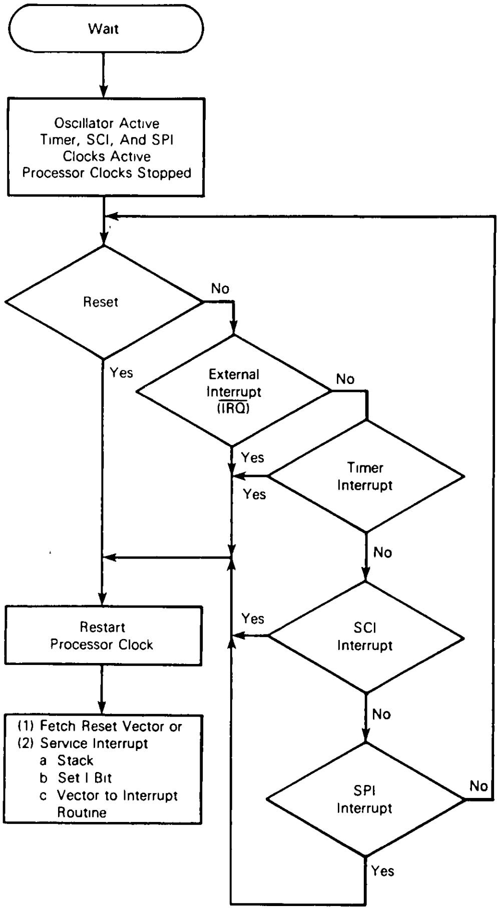

Wait
Oscillator Active
Timer, SCI, And SPI
Clocks Active
Processor Clocks Stopped
Reset
No
Yes
External
Interrupt
($\overline{IRQ}$)
No
Yes
Timer
Interrupt
Yes
No
Restart
Processor Clock
Yes
SCI
Interrupt
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt
Routine
SPI
Interrupt
No
Yes

Figure 10. WAIT Function Flowchart

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, \$18–\$19 (counter register) or

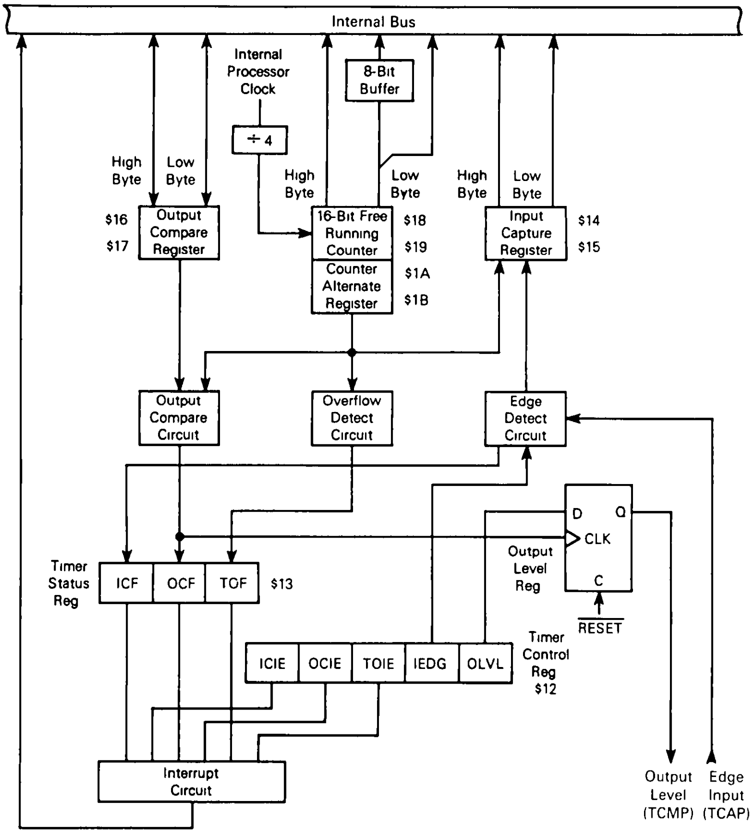

Internal Bus
Internal Processor Clock
÷4
8-Bit Buffer
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
$18
$19
Counter Alternate Register
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
Timer Control Reg $12
Interrupt Circuit
Output Level (TCMP)
Edge Input (TCAP)

Figure 11. Timer Block Diagram

$1A–$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

IEDG — Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
1 = Positive edge
0 = Negative edge
Reset does not affect te IEDG bit (U = unaffected).

OLVL — Output Level
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
1 = High output
0 = Low output

Bits 2, 3, and 4 — Not used
Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF — Output Compare Flag
1 = Flag set when output compare register contents match the free-running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address \$1A and \$1B contains the same value as the free-running counter (at address \$18 and \$19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to \$FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 12.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

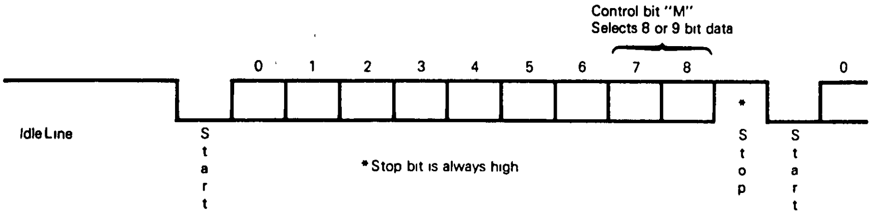


Figure 12. Data Format

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 13); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 13. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates, which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

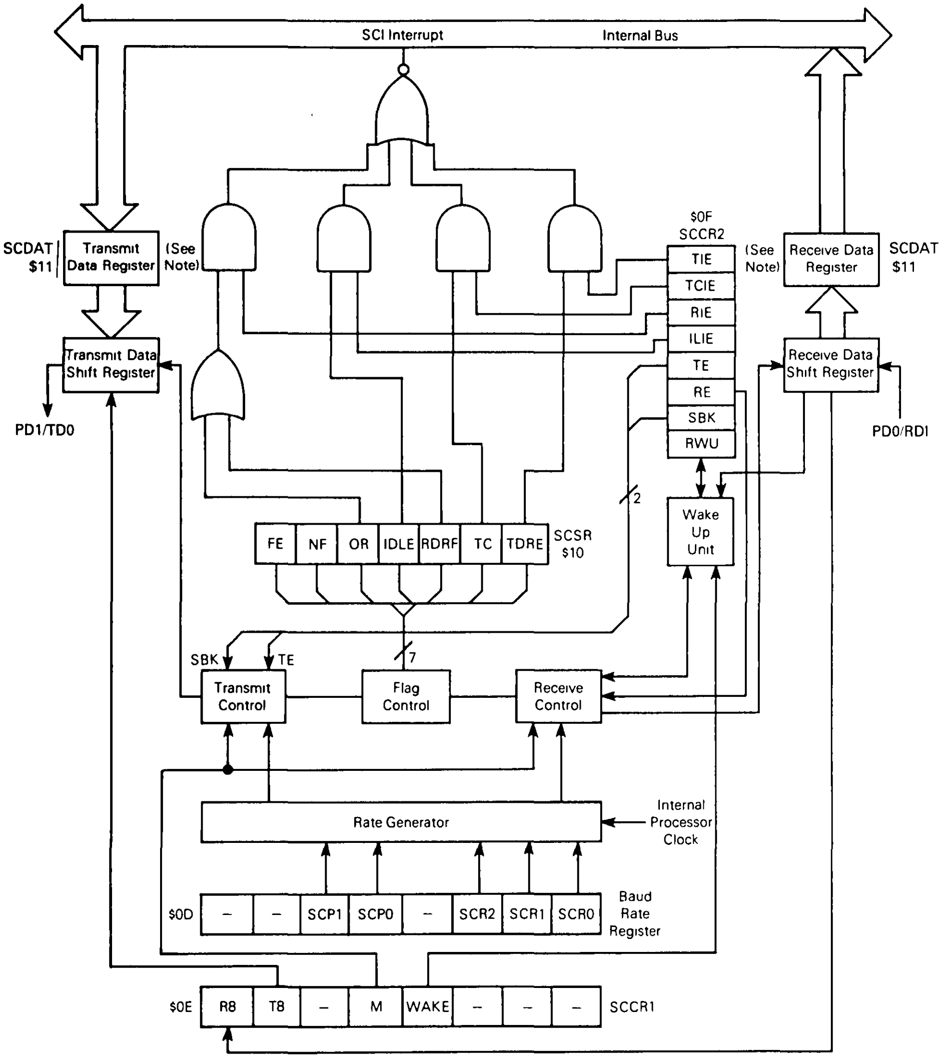
	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET:	U	U	U	U	U	U	U	U

As shown in Figure 13, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $0E

The SCCR1 provides control bits that determine word length and select the wake-up method.

	7	6	5	4	3	2	1	0
	R8	T8	—	M	WAKE	—	—	—
RESET:	U	U	—	U	U	—	—	—


SCI Interrupt
Internal Bus
SCDAT $11
Transmit Data Register
(See Note)
Transmit Data Shift Register
PD1/TD0
$0F SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(See Note)
Receive Data Register
SCDAT $11
Receive Data Shift Register
PD0/RDI
Wake Up Unit
2
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $10
SBK
TE
7
Transmit Control
Flag Control
Receive Control
Rate Generator
Internal Processor Clock
$0D
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
Baud Rate Register
$0E
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 13. SCI Block Diagram

R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most significant bit)
0 = Idle line condition

Bits 0–2, and 5 — Not used
Can read either one or zero

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $OF

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

RESET·

0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
- 1 = SCI interrupt enabled
- 0 = TDRE interrupt disabled

TCIE — Transmit Complete Interrupt Enable
- 1 = SCI interrupt enabled
- 0 = TC interrupt disabled

RIE — Receive Interrupt Enable
- 1 = SCI interrupt enabled
- 0 = RDRF and OR interrupts disabled

ILIE — Idle Line Interrut Enable
- 1 = SCI interrupt enabled
- 0 = Idle interrupt disabled

TE — Transmit Enable
- 1 = Transmit shift register output is applied to the TD0 line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
- 0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TD0 line becomes a high-impedance line.

RE — Receive Enable
- 1 = Receiver shift register input is applied to the RDI line.
- 0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.

RWU — Receiver Wake-Up
- 1 = Places receiver in sleep mode and enables wake-up function
- 0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
  Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)

SBK — Send Break
- 1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
- 0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	—

RESET

1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
- 1 = TDR contents transferred to the transmit data shift register
- 0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1), followed by a write to the TDR.

TC — Transmit Complete
- 1 = Indicates end of data frame, preamble, or break condition has occurred
- 0 = TC bit cleared by reading the SCSR (with TC = 1), followed by a write to the TDR

RDRF — Receive Data Register (RDR) Full
- 1 = Receive data shift register contents transferred to the RDR
- 0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR

IDLE — Idle Line Detect
- 1 = Indicates receiver has detected an idle line
- 0 = IDLE is cleared by reading the SCSR (with IDLE = 1), followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error
- 1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
- 0 = OR is cleared by reading the SCSR (with OR = 1), followed by a read of the RDR.

NF — Noise Flag
- 1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
- 0 = NF is cleared by reading the SCSR (with NF = 1), followed by a read of the RDR.

FE — Framing Error
- 1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.

0 = NF is cleared by reading the SCSR (with FE = 1), followed by a read of the RDR.

Bit 0 — Not used

Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

	7	6	5	4	3	2	1	0
	—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET:	—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0

SCP1 — SCI Prescaler Bit 1

Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 3.

SCR0 — SCI Baud Rate Bit 0

SCR1 — SCI Baud Rate Bit 1

SCR2 — SCI Baud Rate Bit 2

Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 4.

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided-by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600-Hz baud rate is required with a 2.4576-MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 14) or MCUs that can be either masters or slaves.

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	0		4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25 60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15 625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19 20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

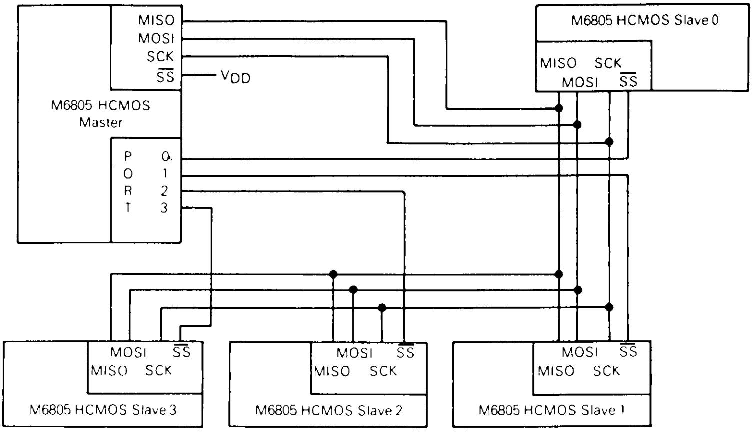

MISO
MOSI
SCK
SS
VDD
M6805 HCMOS Master
P O R T
0
1
2
3
M6805 HCMOS Slave 0
MISO SCK
MOSI SS
MOSI SS
MISO SCK
M6805 HCMOS Slave 3
M6805 HCMOS Slave 2
M6805 HCMOS Slave 1

Figure 14. Master-Slave System Configuration

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS} = 1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 15, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

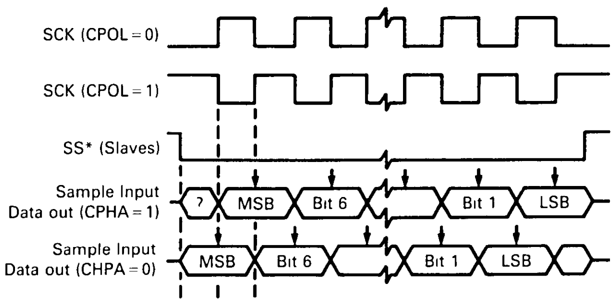

SCK (CPOL = 0)
SCK (CPOL = 1)
SS* (Slaves)
Sample Input
Data out (CPHA = 1)
?
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA = 0)
MSB
Bit 6
Bit 1
LSB

Figure 15. Data Clock Timing Diagram

Slave Select

The slave select ($\overline{\text{SS}}$) input line selects a slave device. The $\overline{\text{SS}}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{\text{SS}}$ line on the master must be tied high; if the $\overline{\text{SS}}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{\text{SS}}$ with SCK. In this clock phase mode, $\overline{\text{SS}}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{\text{SS}}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{\text{SS}}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{\text{SS}}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 16. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the $\overline{\text{SS}}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 17 illustrates the MOSI, MISO, SCK, and $\overline{\text{SS}}$ master-slave interconnections.

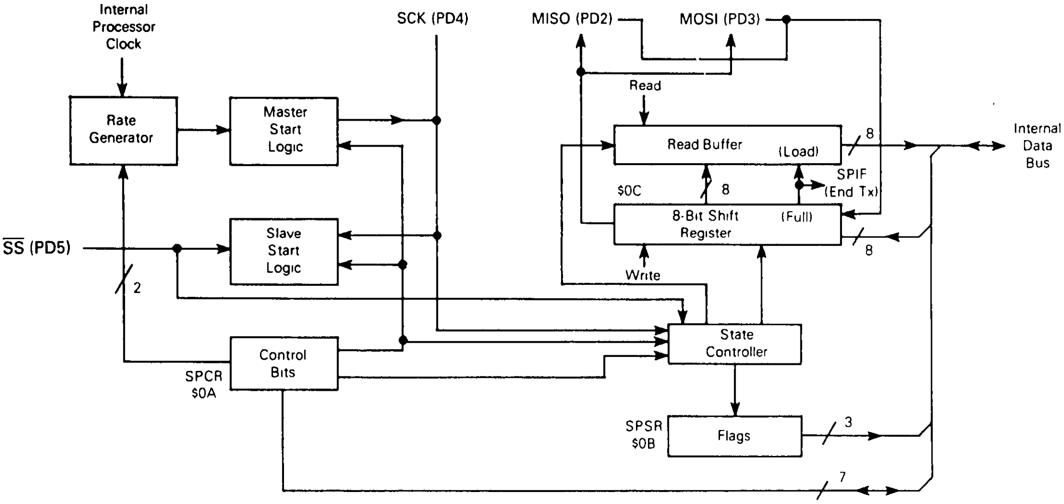

Internal Processor Clock
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Rate Generator
Master Start Logic
Read
Read Buffer
(Load)
8
Internal Data Bus
$0C
8
SPIF (End Tx)
8-Bit Shift Register
(Full)
8
$\overline{\text{SS}}$ (PD5)
Slave Start Logic
Write
2
State Controller
Control Bits
SPCR $0A
SPSR $0B
Flags
3
7

Figure 16. SPI Block Diagram

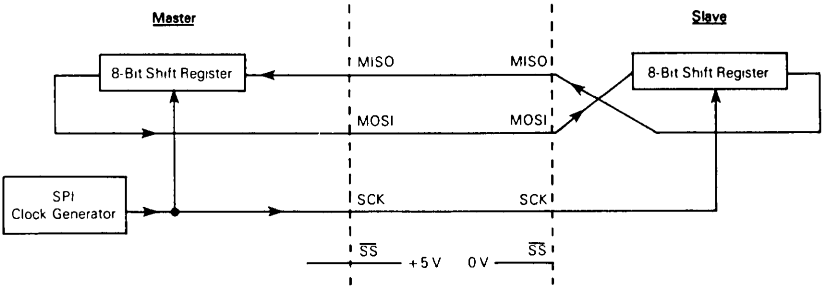

Master
Slave
8-Bit Shift Register
MISO
MISO
8-Bit Shift Register
MOSI
MOSI
SPI Clock Generator
SCK
SCK
$\overline{\text{SS}}$
+5 V
0 V
$\overline{\text{SS}}$

Figure 17. SPI Master-Slave Interconnections

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

	7	6	5	4	3	2	1	0
	SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity

Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase

Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits

Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 – Not used

Can read either one or zero

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

	7	6	5	4	3	2	1	0
	SPIF	WCOL	—	MODF	—	—	—	—
RESET	0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device.
(If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.

WCOL — Write Collision
1 = Indicates an attempt is made to write to SPDR while data transfer is in process.
0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict.
0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 0–3, and 5 — Not used

Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read/write register used to receive and transmit SPI data.

	7	6	5	4	3	2	1	0
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET	U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register

— Continued —

Operation	X:A X*A			
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ

— Continued —

Function	Mnemonic
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such,

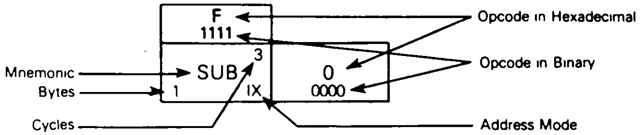
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Low \ Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi \ Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to 2×V_{DD}+0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HCL05C8P, FN	T_A	T_L to T_H 0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		60	
Plastic Leaded Chip Carrier (PLCC)		70	

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

$V_{DD} = 4.5$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0, PD5, PD7	1.9 kΩ	2.26 kΩ	200 pF

$V_{DD} = 3.0$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ 6.32 kΩ	50 pF	
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF

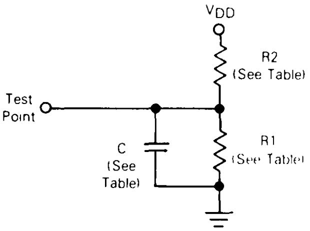


Figure 18. Equivalent Test Load

DC ELECTRICAL CHARACTERISTICS

($V_{DD} = 5.0$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage ($I_{Load} = 0.8$ mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) ($I_{Load} = 1.6$ mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 21) ($I_{Load} = 1.6$ mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 23) Wait (see Figures 22 and 23) Stop (see Figure 23) 25°C 0° to 70°C	I_{DD}	 — — — —	 — — — —	 5.0 2.75 15 25	 mA mA μA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±1.0	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}. Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source ($f_{osc} = 2.1$ MHz), all inputs 0.2 V from rail, no dc loads, less than 50 pF on all outputs, $C_L = 20$ pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, $V_{IL} = 0.2$ V, $V_{IH} = V_{DD} - 0.2$ V
6. Stop I_{DD} measured with OSC1 = V_{SS}
7. Standard temperature range is 0° to 70°C. A 25°C only version is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.4 Vdc–3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ μA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 0.4 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.3$ $V_{DD} - 0.3$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (3.6 Vdc at 1.0 MHz) Run (See Figures 22 and 24) Wait (See Figures 22 and 24) Stop (See Figure 24) 25°C 0°C to 70°C	I_{DD}	 — — — —	 — — — —	 1.75 900 5.0 10	 mA μA μA μA
Supply Current (2.4 Vdc at 500 kHz) Run (See Figures 22 and 25) Wait (See Figures 22 and 25) Stop (See Figure 25) 25°C 0° to 70°C	I_{DD}	 — — — —	 — — — —	 750 400 2.0 5.0	 μA μA μA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±1.0	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

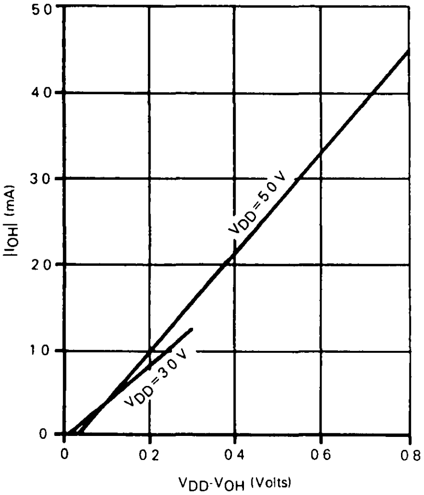


Figure 19. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

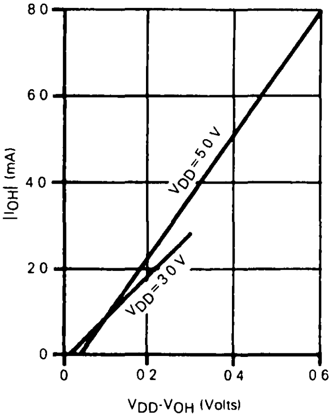


Figure 20. Typical V_{OH} vs I_{OH} for PD1-PD4

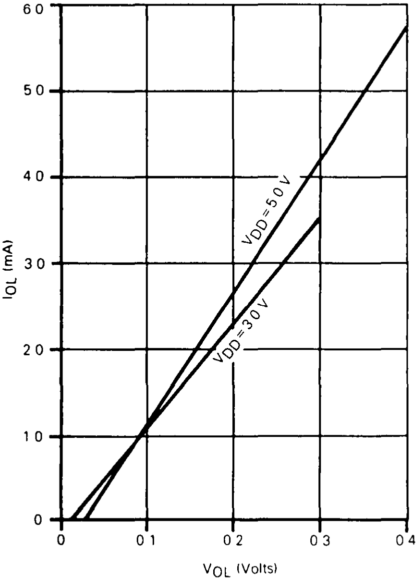


Figure 21. Typical V_{OL} vs I_{OL} for All Ports

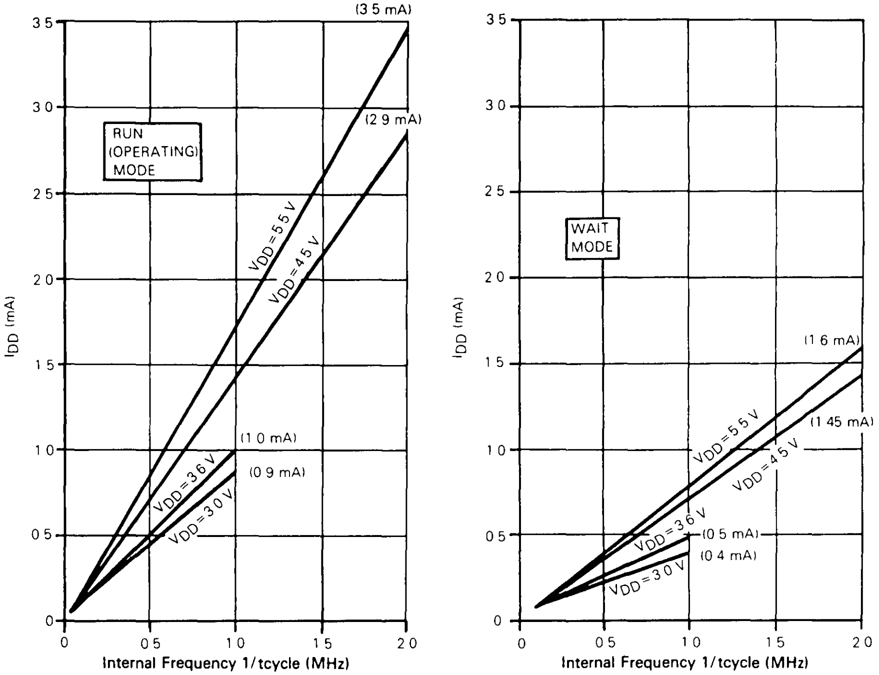


Figure 22. Typical Current vs Internal Frequency for Run and Wait Modes

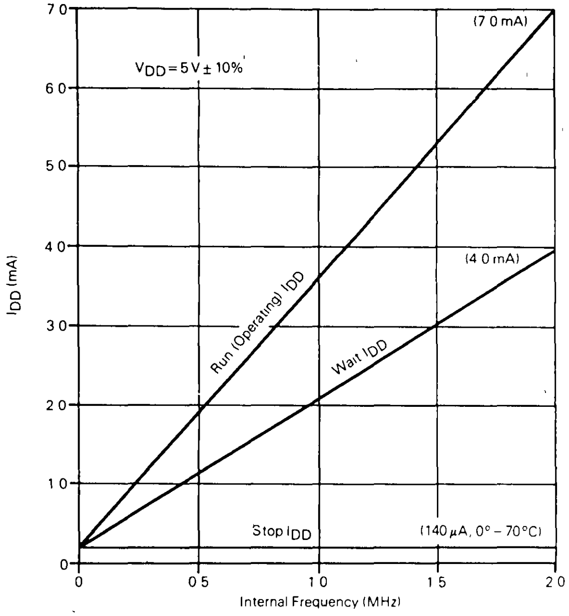


Figure 23. Maximum I_{DD} vs Frequency for $V_{DD}=5.0$ Vdc

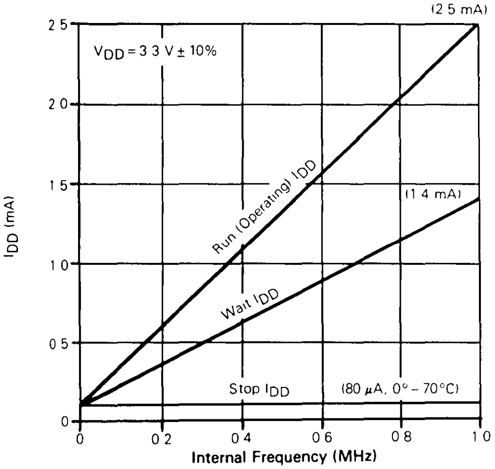


Figure 24. Maximum I_{DD} vs Frequency for $V_{DD}=3.3$ Vdc

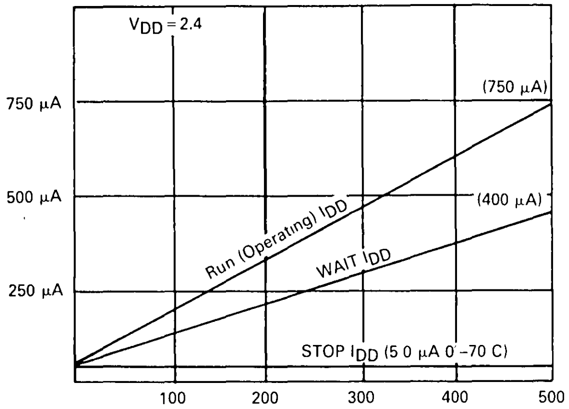


Figure 25. Maximum I_{DD} vs Frequency for $V_{DD}=2.4$ Vdc

CONTROL TIMING

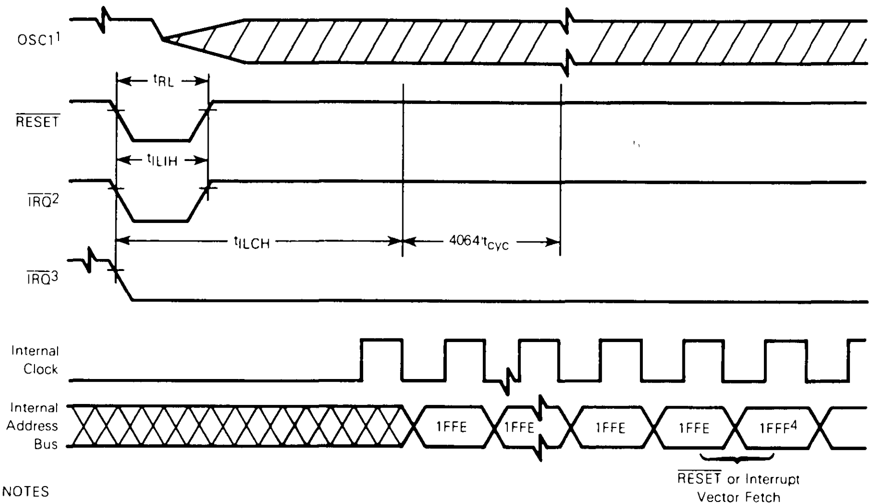
($V_{DD} = 5.0$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} - 2$)		—	2.1	
External Clock ($f_{osc} - 2$)		dc	2.1	
Cycle Time (see Figure 28)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 27)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 27)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3 $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option
4 $\overline{\text{RESET}}$ vector address shown for timing example

Figure 26. Stop Recovery Timing Diagram

CONTROL TIMING

($V_{DD} = 3.3$ Vdc ± 0.3 Vdc, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} \div 2$)		—	1.0	
External Clock ($f_{osc} \div 2$)		dc	1.0	
Cycle Time (see Figure 29)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 29)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 29)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 27)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 27)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

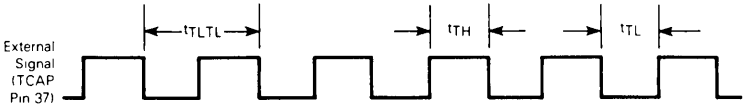


Figure 27. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

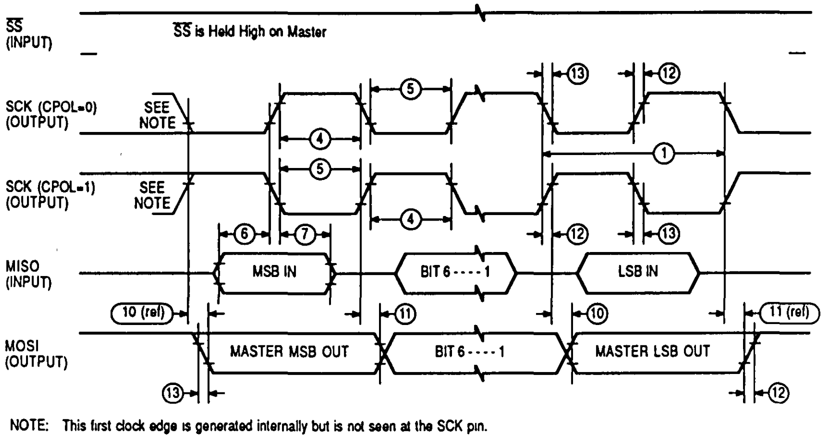
SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 1.0	 — —	 t_{cyc} μs
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 500	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 500	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 720 400	 — —	 μs ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 720 400	 — —	 μs ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 200 200	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 200 200	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 250	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 500	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0 25 —	 — 500	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 200 2.0	 ns μs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 200 2.0	 ns μs

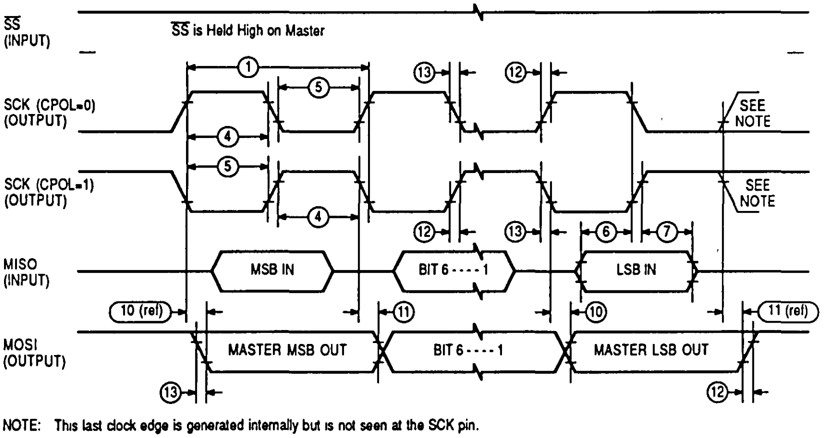
*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
10 (ref)
⑩
⑪
11 (ref)
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

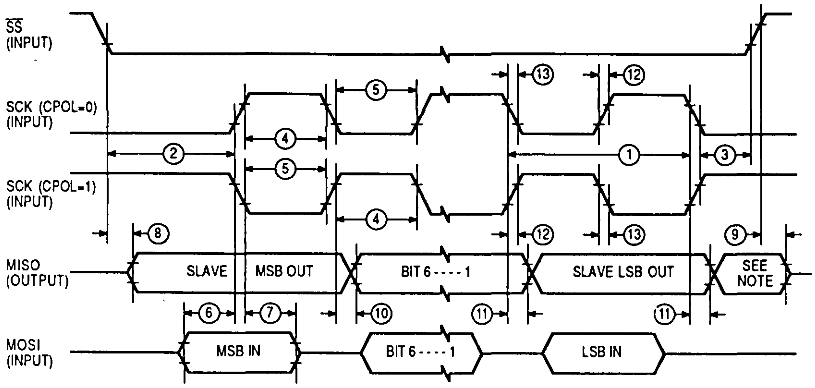
a) SPI MASTER TIMING (CPHA = 0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
10 (ref)
⑩
⑪
11 (ref)
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

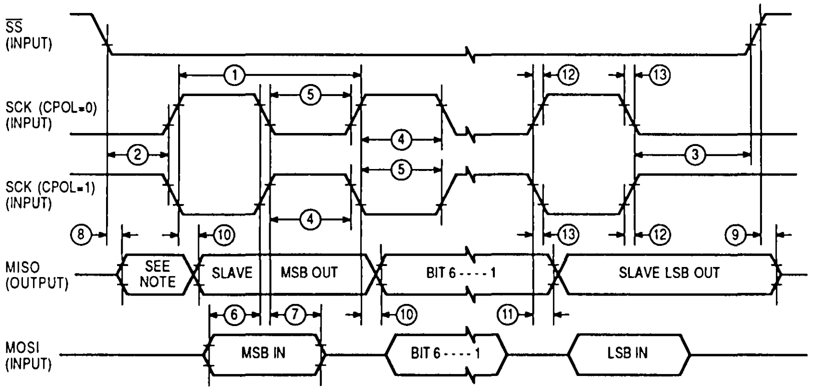
b) SPI MASTER TIMING (CPHA = 1)

Figure 28. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE: Not defined but normally MSB of character just received

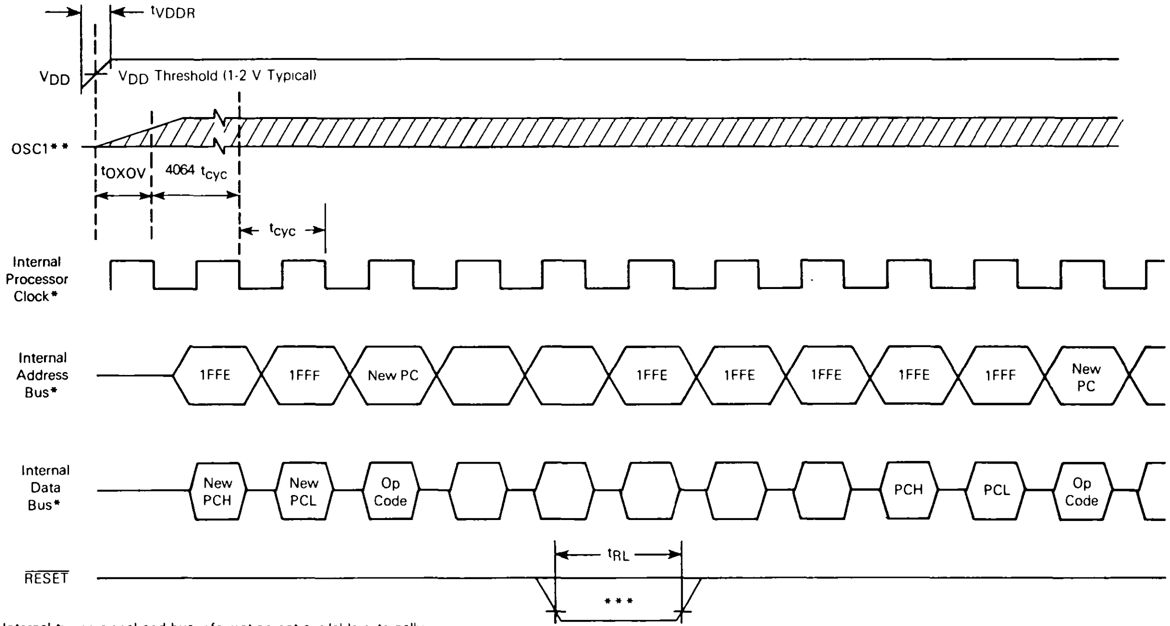
c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA = 1)

Figure 28. SPI Timing Diagrams (Sheet 2 of 2)


t_{VDDR}
V_{DD}
V_{DD} Threshold (1-2 V Typical)
OSC1**
t_{OXOV}
4064 t_{cyc}
t_{cyc}
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
t_{RL}
RESET
***

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency. It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 29. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS™, disk file
MS™-DOS/PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM MC68HC805C4

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS/PC-DOS Disk File

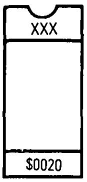
MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 68766 EPROM device, the EPROM must be programmed as described in the following paragraph.

Start the page zero, user ROM at EEPROM address \$0020 through \$004F. Start the user ROM at EEPROM address \$0100 through \$1EFF with vectors from \$1FF4 to \$1FFF. All unused bytes, including the user's space, must be set to zero.

To use a 2764 or 6874 EPROM or the EEPROM in an MC68HC805C4, two are required. Start the page zero user ROM data at EPROM or EEPROM address \$0020 through \$004F in the first device. Start the user ROM data at address \$0100 through\$10FF in the first device. The remainder of the user ROM data should go from \$0100 through \$10FF in the second device, with vectors from \$0004 through \$000F. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

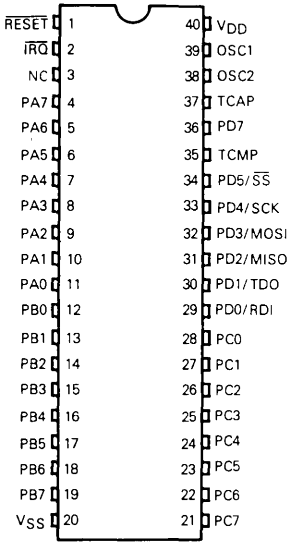
The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HCL05C8 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C	MC68HCL05C8P
PLCC (FN Suffix)	0°C to +70°C	MC68HCL05C8FN

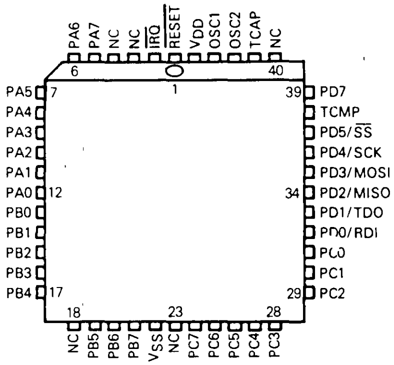
MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE


RESET 1
IRQ 2
NC 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE


PA6
PA7
NC
NC
IRQ
RESET
VDD
OSC1
OSC2
TCAP
NC
6
40
1
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
VSS
NC
PC7
PC6
PC5
PC4
PC3

NOTE Bulk substrate tied to V_{SS}.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HSC05C4

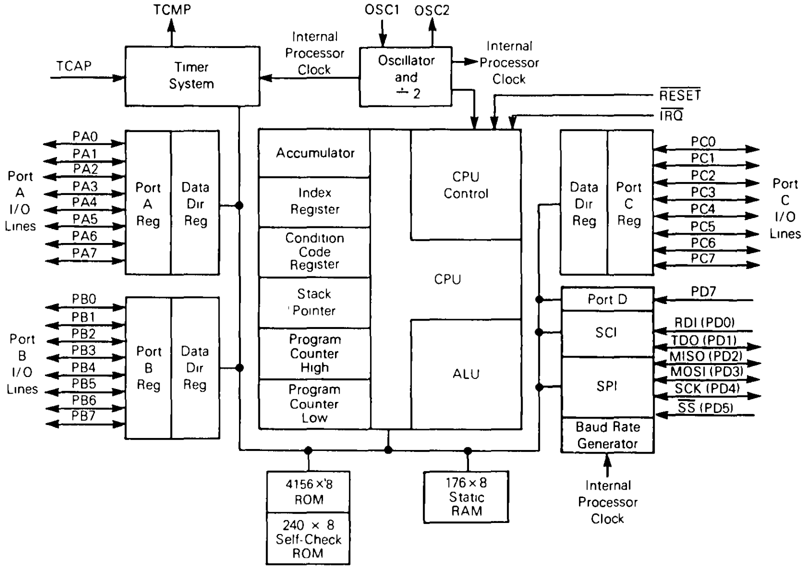
Technical Summary
8-Bit Microcontroller Unit

The MC68HSC05C4 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 4156 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8×8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
OSC1
OSC2
TCAP
Timer System
Internal Processor Clock
Oscillator and ÷ 2
Internal Processor Clock
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I/O Lines
Port A Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I/O Lines
Port B Reg
Data Dir Reg
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
SS (PD5)
Baud Rate Generator
Internal Processor Clock
4156×8 ROM
240 × 8 Self-Check ROM
176×8 Static RAM

This document contains information on a new product Specifications and information herein are subject to change without notice

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{OSC} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

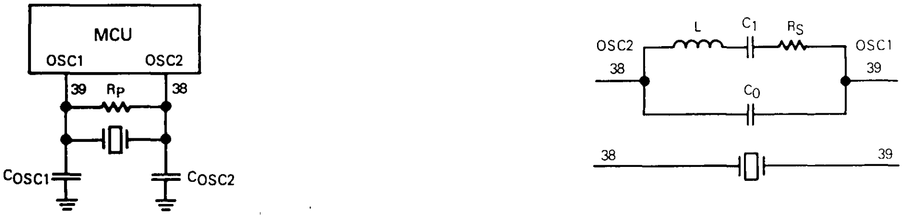
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

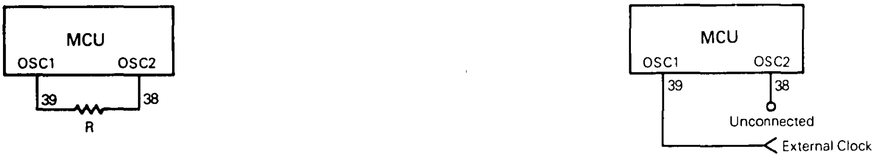
	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

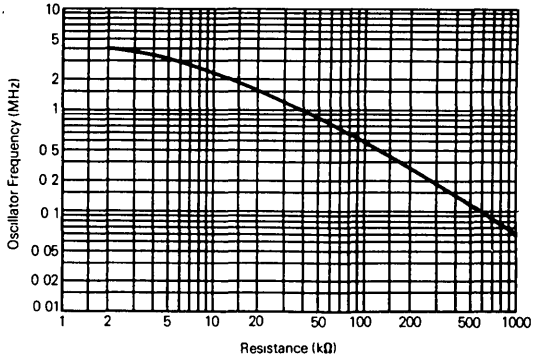


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

$R/\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*$R/\overline{W}$ is an internal signal.

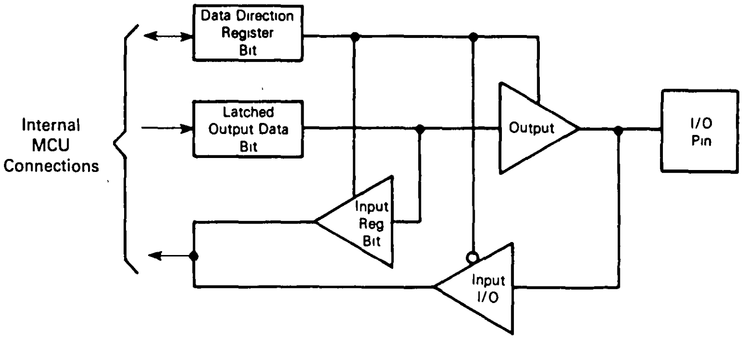


Figure 3. Typical Port I/O Circuit

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

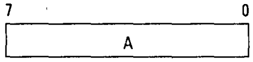
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

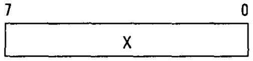
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



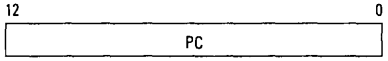
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0.

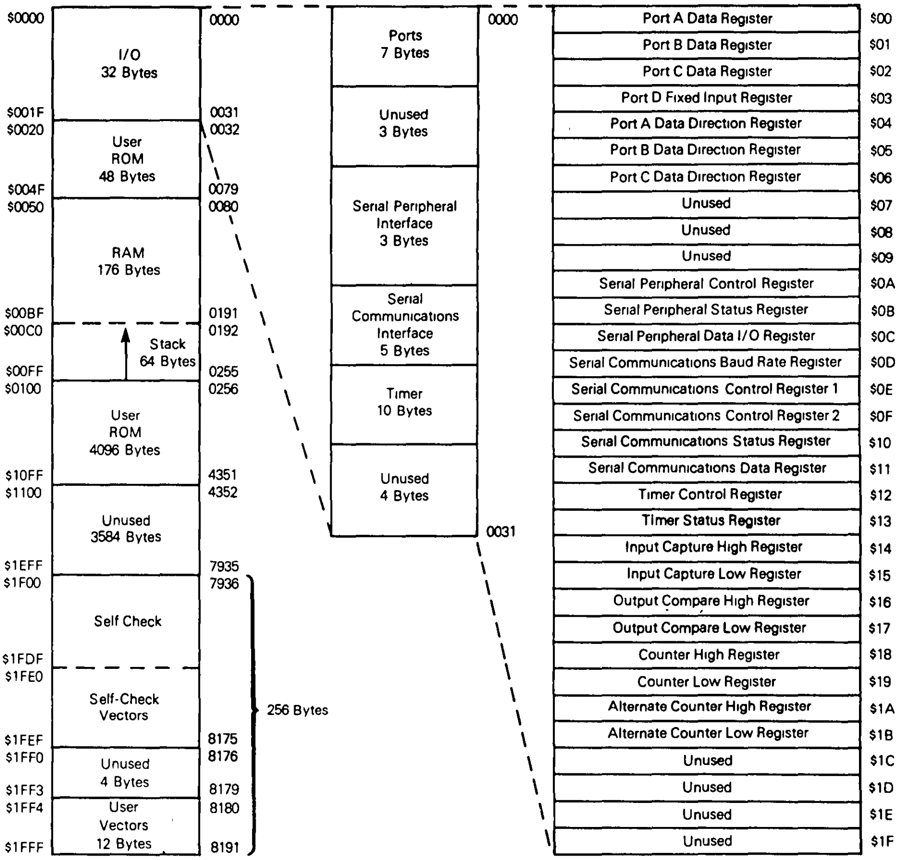

$0000
I/O
32 Bytes
$001F
$0020
User
ROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
User
ROM
4096 Bytes
$10FF
$1100
Unused
3584 Bytes
$1EFF
$1F00
Self Check
$1FDF
$1FE0
Self-Check
Vectors
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
4351
4352
7935
7936
8175
8176
8179
8180
8191
256 Bytes
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Unused
4 Bytes
0000
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Unused $1C
Unused $1D
Unused $1E
Unused $1F

Figure 4. Memory Map

Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

12					7		0
0	0	0	0	0	1	1	SP

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

CCR

H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

- I/O — Exercise of ports A, B, and C
- RAM — Counter test for each RAM byte
- ROM — Exclusive OR with odd ones parity result
- Timer — Tracks counter register and checks OCF flag
- Interrupts — Tests external, timer, SCI and SPI interrupts
- SCI — Transmission test; checks RDRF, TDRE, TC, and FE flags
- SPI — Transmission test; checks SPIF, WCOL, and MODF flags

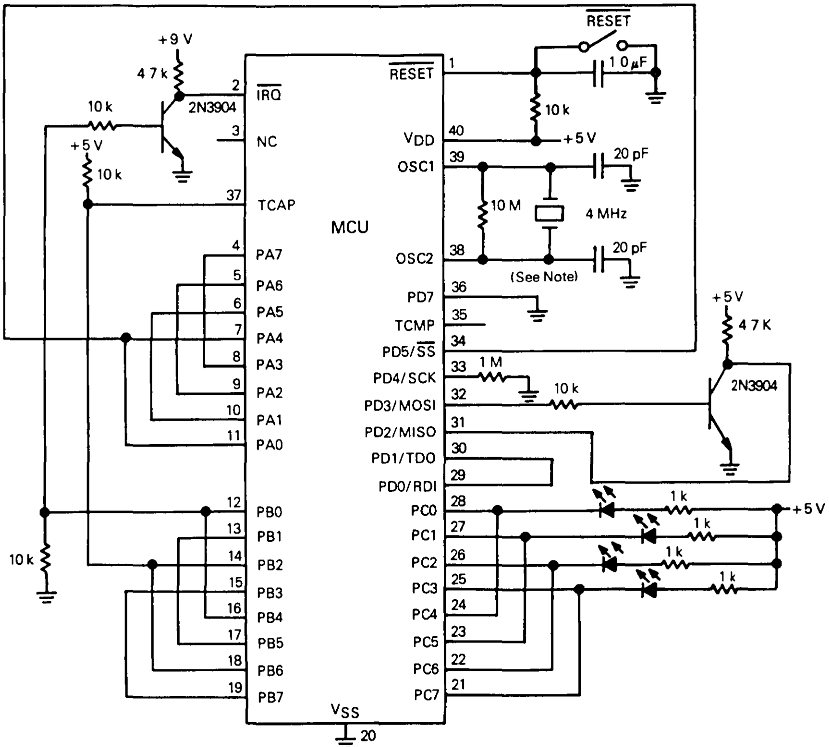
Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM



+9 V
4.7 k
10 k
2N3904
+5 V
10 k
2
3
37
IRQ
NC
TCAP
4
5
6
7
8
9
10
11
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
12
13
14
15
16
17
18
19
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
10 k
MCU
VSS
20
RESET
RESET
1
1.0 μF
10 k
VDD
40
+5 V
OSC1
39
20 pF
10 M
4 MHz
OSC2
38
20 pF
(See Note)
PD7
36
TCMP
35
PD5/SS
34
PD4/SCK
33
1 M
PD3/MOSI
32
10 k
PD2/MISO
31
PD1/TDO
30
PD0/RDI
29
+5 V
4.7 K
2N3904
PC0
28
PC1
27
PC2
26
PC3
25
PC4
24
PC5
23
PC6
22
PC7
21
1 k
1 k
1 k
1 k
+5 V

NOTE The RC Oscillator Option may also be used in this circuit

Figure 5. Self-Check Circuit Schematic Diagram

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or IRQ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on; 1 indicates LED is off.

checksum subroutine is called at location \$1F93 with RAM location \$0053 equal to \$01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations \$0050 through \$0053 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input (RESET). The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the RESET pin is low at the end of 4064 t_{cyc}, the MCU will remain in the reset condition until RESET goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts (IRQ, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

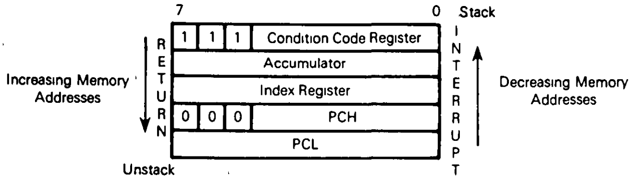
If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request



NOTE Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc Pulling from the stack is in the reverse order

Figure 6. Interrupt Stacking Order

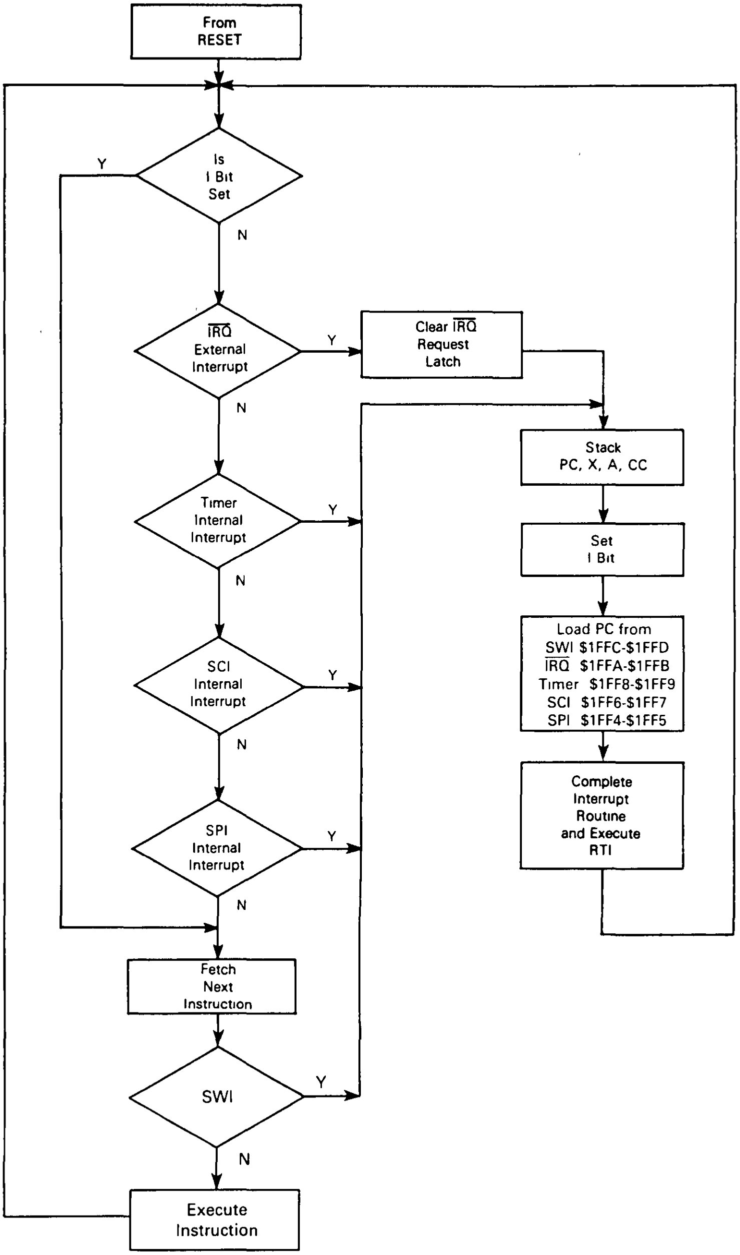

From RESET
Is I Bit Set
Y
N
IRQ External Interrupt
Y
Clear IRQ Request Latch
N
Timer Internal Interrupt
Y
N
SCI Internal Interrupt
Y
N
SPI Internal Interrupt
Y
N
Fetch Next Instruction
SWI
Y
N
Execute Instruction
Stack PC, X, A, CC
Set I Bit
Load PC from
SWI $1FFC-$1FFD
IRQ $1FFA-$1FFB
Timer $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI $1FF4-$1FF5
Complete Interrupt Routine and Execute RTI

Figure 7. Reset and Interrupt Processing Flowchart

input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of \$1FFA and \$1FFB.

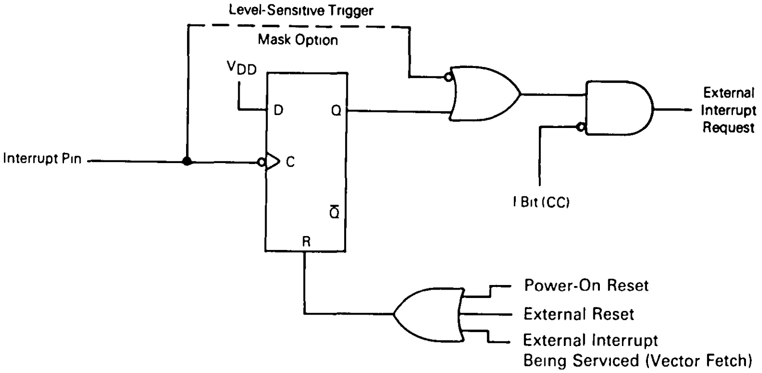
Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

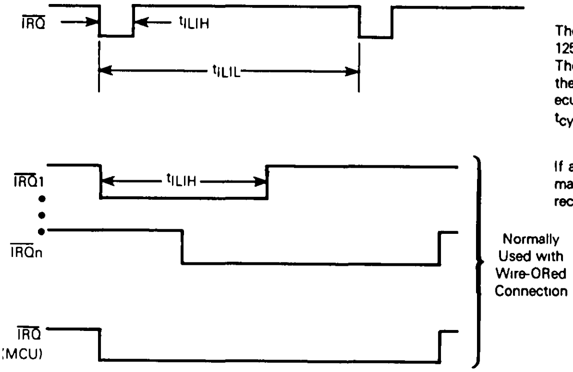
The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI


Level-Sensitive Trigger
Mask Option
VDD
D
Q
C
Q̄
R
Interrupt Pin
External Interrupt Request
I Bit (CC)
Power-On Reset
External Reset
External Interrupt Being Serviced (Vector Fetch)

(a) Interrupt Internal Function Diagram


IRQ
tILIH
tILIL
IRQ1
IRQn
IRQ (MCU)
Normally Used with Wire-ORed Connection

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 8. External Interrupt

operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI during STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the IRQ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the IRQ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the IRQ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active (refer to Figure 10). An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer

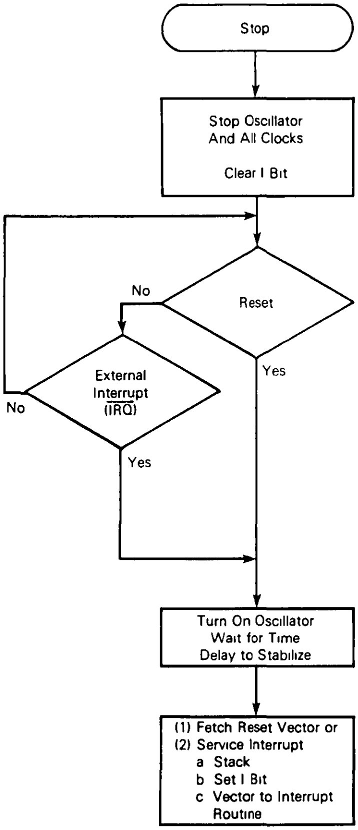

Stop
Stop Oscillator And All Clocks
Clear I Bit
Reset
No
Yes
External Interrupt (IRQ)
No
Yes
Turn On Oscillator Wait for Time Delay to Stabilize
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt Routine

Figure 9. STOP Function Flowchart

may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{\text{RESET}}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

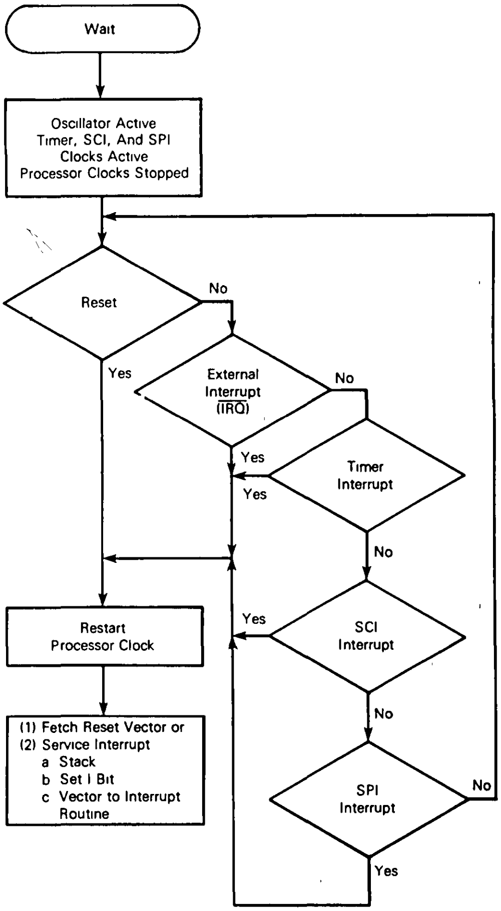

Wait
Oscillator Active Timer, SCI, And SPI Clocks Active Processor Clocks Stopped
Reset
No
Yes
External Interrupt ($\overline{\text{IRQ}}$)
No
Yes
Timer Interrupt
Yes
No
Restart Processor Clock
Yes
SCI Interrupt
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt Routine
SPI Interrupt
No
Yes

Figure 10. WAIT Function Flowchart

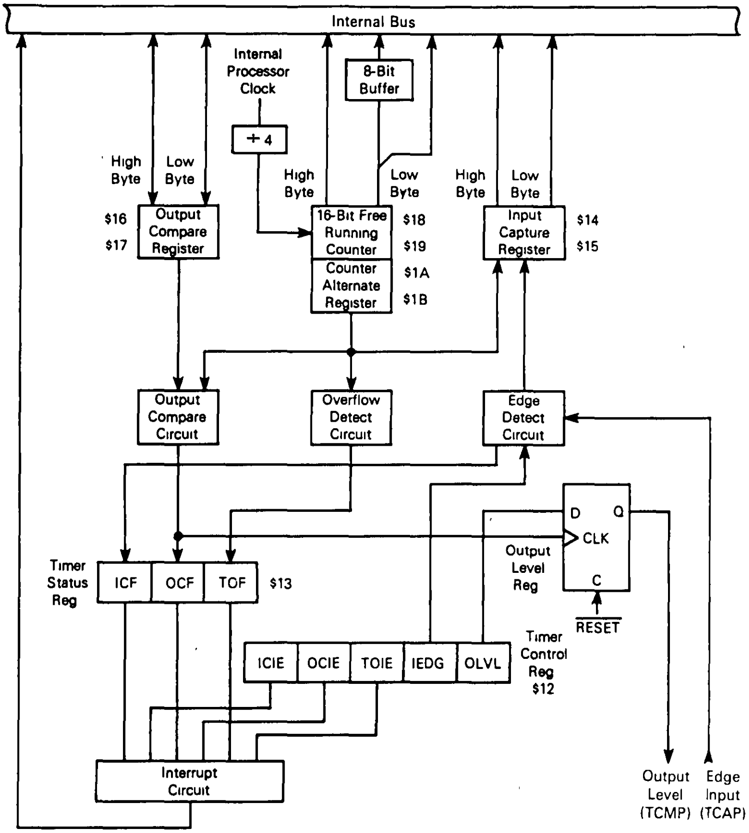

Internal Bus
Internal Processor Clock
8-Bit Buffer
÷ 4
High Byte
Low Byte
High Byte
Low Byte
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
$18
$19
Counter Alternate Register
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
Timer Control Reg $12
Interrupt Circuit
Output Level (TCMP)
Edge Input (TCAP)

Figure 11. Timer Block Diagram

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18–$19 (counter register) or $1A–$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set.

An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET.	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

IEDG — Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG bit (U = unaffected).

OLVL — Output Level
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
1 = High output
0 = Low output

Bits 2, 3, and 4 — Not used
Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET.	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF — Output Compare Flag
1 = Flag set when output compare register contents match the free-running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 12.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature

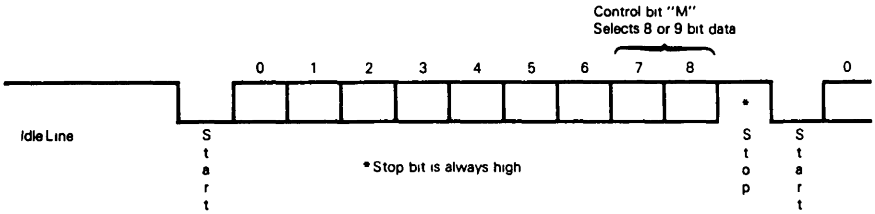


Figure 12. Data Format

is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 13); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF=1, FE=1, receiver data register=$00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 13. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates, which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

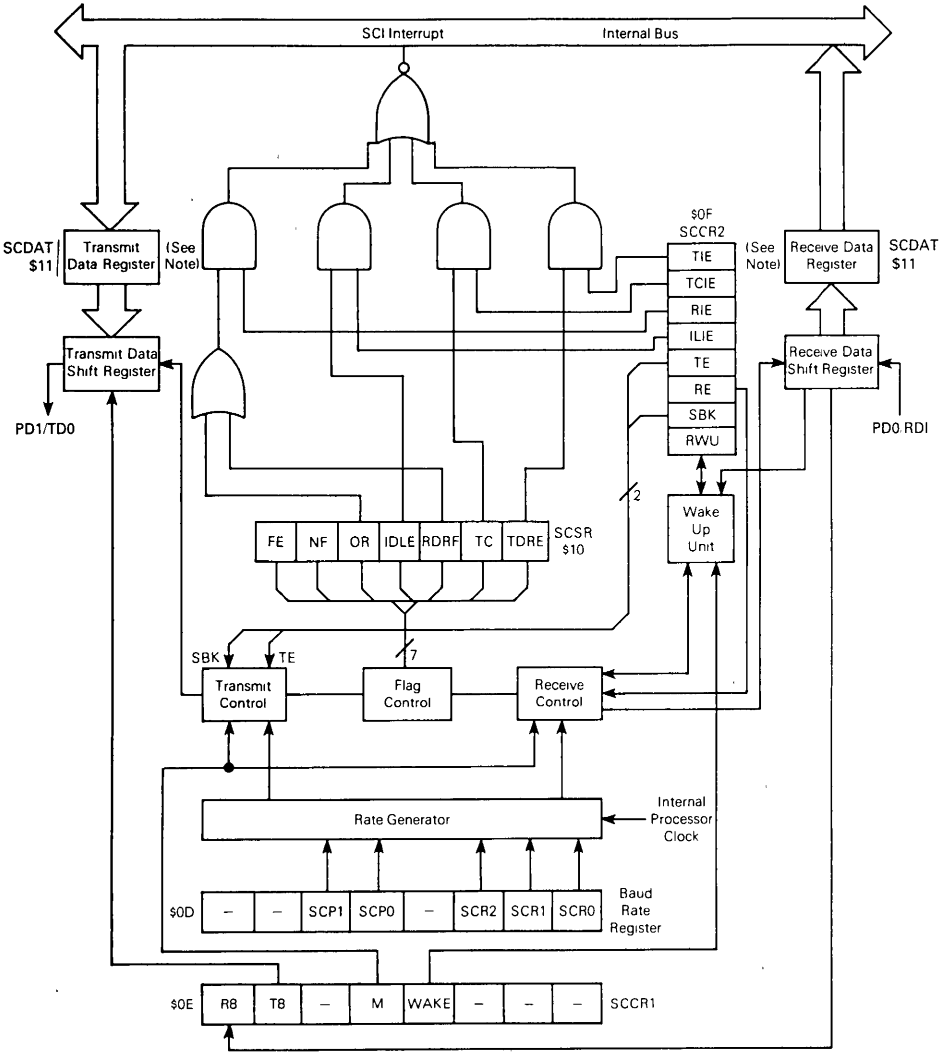
There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET	U	U	U	U	U	U	U	U

As shown in Figure 13, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.


SCI Interrupt
Internal Bus
SCDAT $11
Transmit Data Register
(See Note)
Transmit Data Shift Register
PD1/TD0
$0F SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(See Note)
Receive Data Register
SCDAT $11
Receive Data Shift Register
PD0/RDI
Wake Up Unit
2
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $10
7
SBK
TE
Transmit Control
Flag Control
Receive Control
Rate Generator
Internal Processor Clock
$0D
—
—
SCP1
SCP0
—
SCR2
SCR1
SCR0
Baud Rate Register
$0E
R8
T8
—
M
WAKE
—
—
—
SCCR1

NOTE The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 13. SCI Block Diagram

Serial Communications Control Register 1 (SCCR1) $0E

The SCCR1 provides control bits that determine word length and select the wake-up method.

7	6	5	4	3	2	1	0
R8	T8	—	M	WAKE	—	—	—

RESET:

U U — U U — — —

R8 — Receive Data Bit 8

R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8

T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length

1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

WAKE — Wake-Up Select
  Wake bit selects the receiver wake-up method.
    1 = Address bit (most significant bit)
    0 = Idle line condition
Bits 0–2, and 5 — Not used
  Can read either one or zero

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $0F

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
  1 = SCI interrupt enabled
  0 = TDRE interrupt disabled
TCIE — Transmit Complete Interrupt Enable
  1 = SCI interrupt enabled
  0 = TC interrupt disabled
RIE — Receive Interrupt Enable
  1 = SCI interrupt enabled
  0 = RDRF and OR interrupts disabled
ILIE — Idle Line Interrut Enable
  1 = SCI interrupt enabled
  0 = Idle interrupt disabled
TE — Transmit Enable
  1 = Transmit shift register output is applied to the TD0 line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
  0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TD0 line becomes a high-impedance line.
RE — Receive Enable
  1 = Receiver shift register input is applied to the RDI line.
  0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.
RWU — Receiver Wake-Up
  1 = Places receiver in sleep mode and enables wake-up function
  0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
  Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)
SBK — Send Break
  1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
  0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET	1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
  1 = TDR contents transferred to the transmit data shift register
  0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1), followed by a write to the TDR.
TC — Transmit Complete
  1 = Indicates end of data frame, preamble, or break condition has occurred
  0 = TC bit cleared by reading the SCSR (with TC = 1), followed by a write to the TDR
RDRF — Receive Data Register (RDR) Full
  1 = Receive data shift register contents transferred to the RDR
  0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR
IDLE — Idle Line Detect
  1 = Indicates receiver has detected an idle line
  0 = IDLE is cleared by reading the SCSR (with IDLE = 1), followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.
OR — Overrun Error
  1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
  0 = OR is cleared by reading the SCSR (with OR = 1), followed by a read of the RDR.
NF — Noise Flag
  1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
  0 = NF is cleared by reading the SCSR (with NF = 1), followed by a read of the RDR.

FE — Framing Error

1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.

0 = NF is cleared by reading the SCSR (with FE = 1), followed by a read of the RDR.

Bit 0 — Not used

Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

	7	6	5	4	3	2	1	0
	—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET:	—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0

SCP1 — SCI Prescaler Bit 1

Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 3.

SCR0 — SCI Baud Rate Bit 0

SCR1 — SCI Baud Rate Bit 1

SCR2 — SCI Baud Rate Bit 2

Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 4.

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided-by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600-Hz baud rate is required with a 2.4576-MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz					
1	0		8.0	4.194304	4.0	2.4576	2.0	1.8432
0	0	1	250.000 kHz	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	83.332 kHz	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	62.600 kHz	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	19.200 kHz	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output					
2	1	0		250.000 kHz	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	—	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	125.000 kHz	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	62.500 kHz	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	31.250 kHz	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	15.625 kHz	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	7.813 kHz	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	3.906 kHz	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.953 kHz	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 14) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 2.0 MHz (maximum) master bit frequency
- 4.0 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}$ = 1).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 15, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{SS}$) input line selects a slave device. The $\overline{SS}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high; if the $\overline{SS}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{SS}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

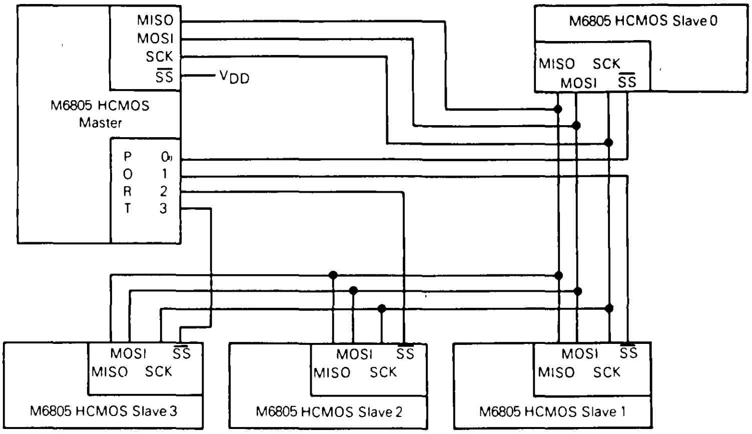


Figure 14. Master-Slave System Configuration

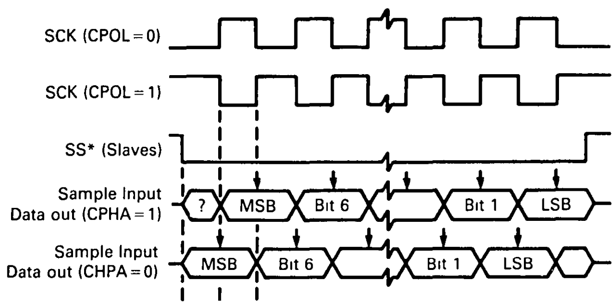

SCK (CPOL=0)
SCK (CPOL=1)
SS* (Slaves)
Sample Input
Data out (CPHA=1)
?
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA=0)
MSB
Bit 6
Bit 1
LSB

Figure 15. Data Clock Timing Diagram

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 16. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 17 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

	7	6	5	4	3	2	1	0
	SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET.								
	0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled

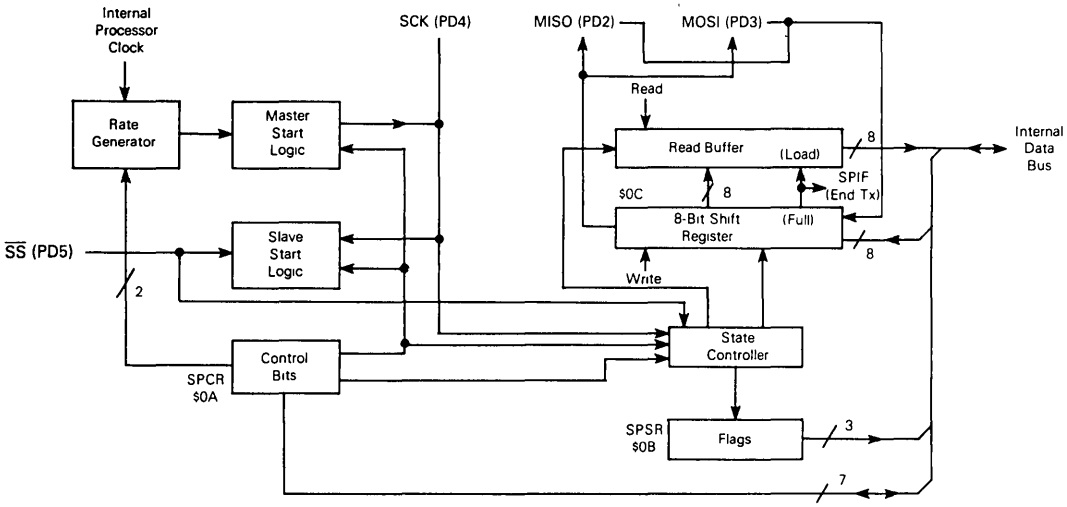

Internal Processor Clock
Rate Generator
Master Start Logic
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Read
Read Buffer
(Load)
8
Internal Data Bus
$0C
8
SPIF (End Tx)
8-Bit Shift Register
(Full)
8
Write
$\overline{SS}$ (PD5)
Slave Start Logic
2
State Controller
Control Bits
SPCR $0A
SPSR $0B
Flags
3
7

Figure 16. SPI Block Diagram

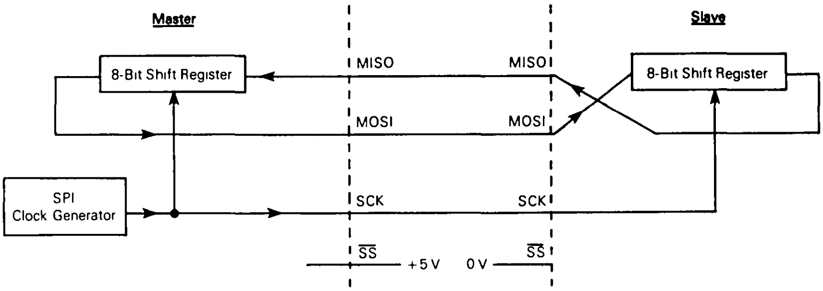


Figure 17. SPI Master-Slave Interconnections

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity

Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase

Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits

Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 – Not used

Can read either one or zero

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—
RESET.							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device.
(If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.

WCOL — Write Collision
1 = Indicates an attempt is made to write to SPDR while data transfer is in process.
0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict.
0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 0–3, and 5 — Not used
Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read/write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET:							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared
Source	MUL
Form(s)	Addressing Mode: Inherent Cycles: 11 Bytes: 1 Opcode: $42

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510

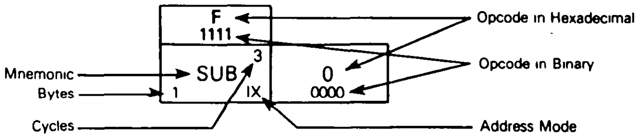
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	$V_{SS}-0.3$ to $2\times V_{DD}+0.3$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		60	
Plastic Leaded Chip Carrier (PLCC)		70	

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

V_{DD} = 4.5 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0,PD5, PD7	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ	6.32 kΩ	50 pF
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF

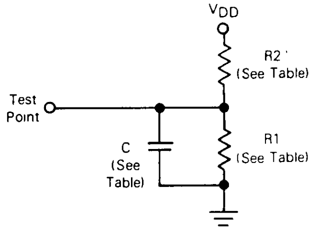


Figure 18. Equivalent Test Load

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 23) Wait (see Figures 22 and 23) Stop (see Figure 23) 25°C 0° to 70°C (Standard)	I_{DD}	 — — — —	 6.7 3.0 2.0 —	 13.3 7.6 50 140	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 8.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD}=3.3 Vdc±10%, V_{SS}=0 Vdc, T_A=T_L to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, I_{Load}≤10.0 µA	V_{OL} V_{OH}	— $V_{DD}-0.1$	— —	0.1 —	V
Output High Voltage (I_{Load}=0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load}=1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD}-0.3$ $V_{DD}-0.3$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load}=0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	0.7×V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	0.2×V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 24) Wait (see Figures 22 and 24) Stop (see Figure 24) 25°C 0° to 70°C (Standard)	I_{DD}	 — — — —	 1.0 0.5 1.0 —	 2.5 1.4 30 80	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE=TE=RE=0). If SPI, SCI active (SPE=TE=RE=1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc}=4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L=20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL}=0.2 V, V_{IH}=$V_{DD}-0.2$ V.
6. Stop I_{DD} measured with OSC1=V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

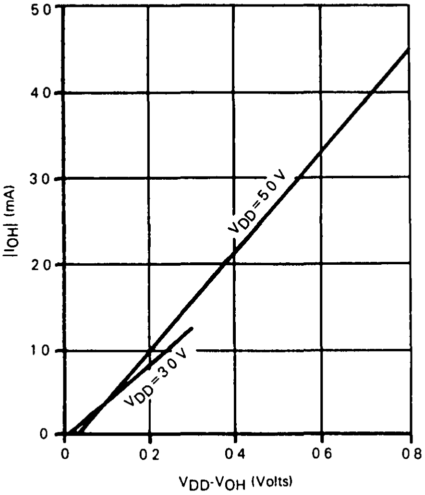


Figure 19. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

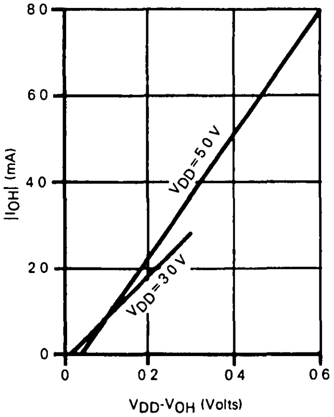


Figure 20. Typical V_{OH} vs I_{OH} for PD1-PD4

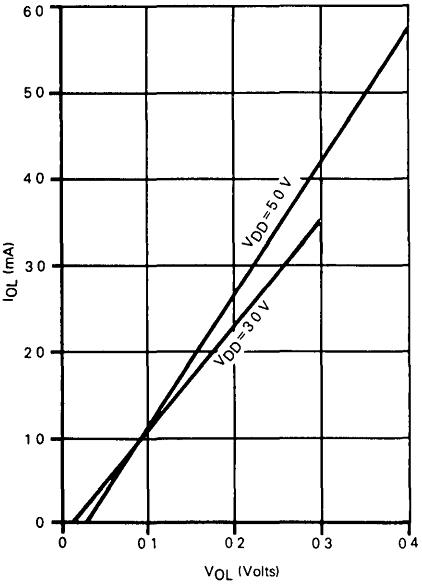


Figure 21. Typical V_{OL} vs I_{OL} for All Ports

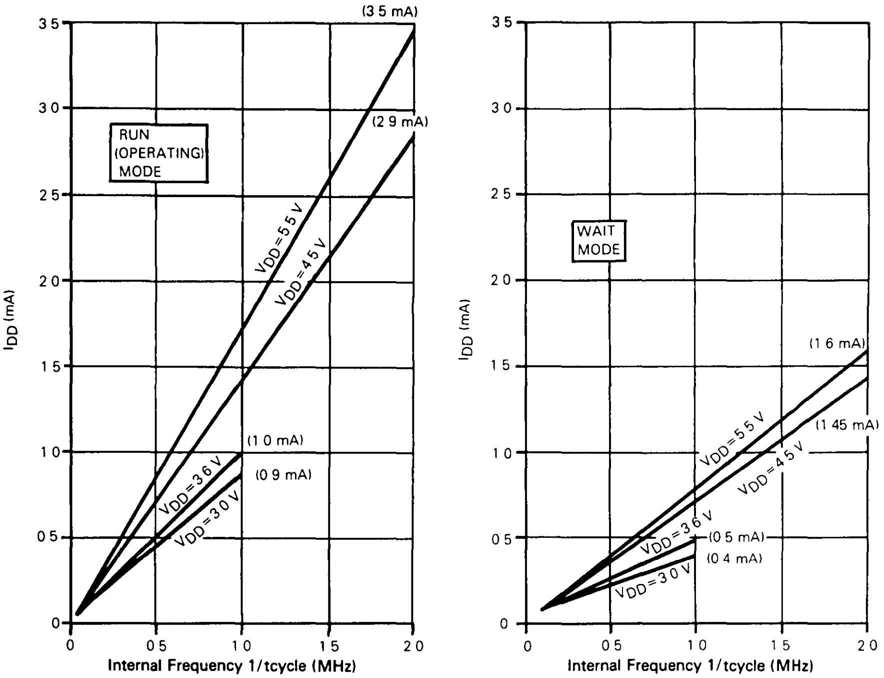


Figure 22. Typical Current vs Internal Frequency for Run and Wait Modes

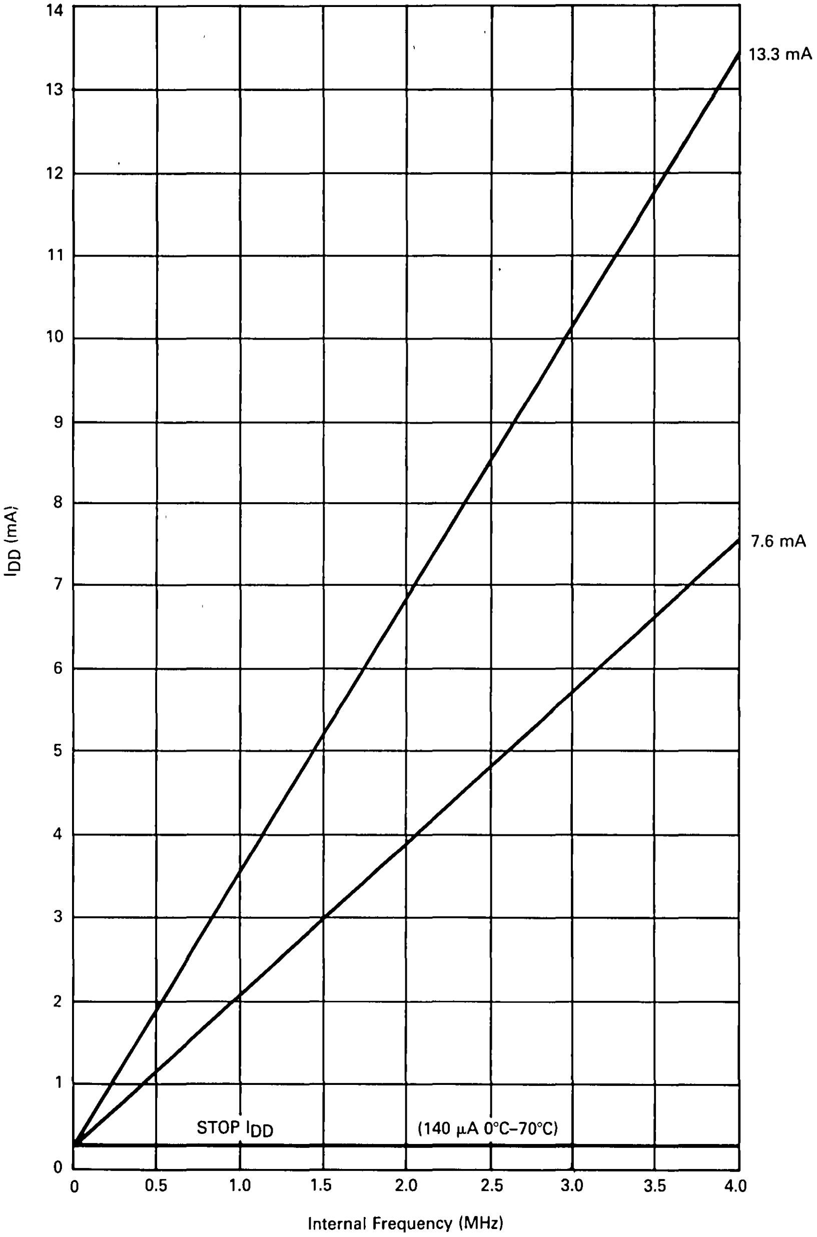


Figure 23. Maximum I_{DD} vs Frequency for $V_{DD}=5.0$ Vdc

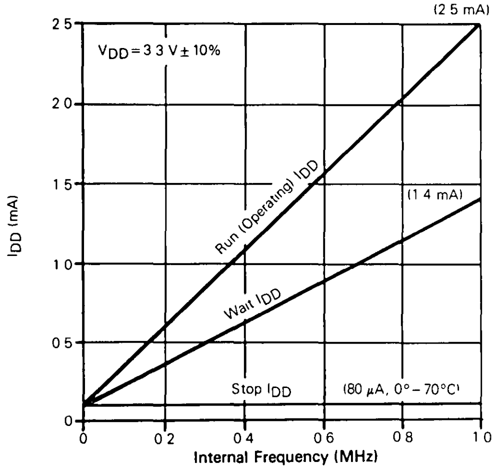


Figure 24. Maximum I_{DD} vs Frequency for $V_{DD}=3.3$ Vdc

CONTROL TIMING

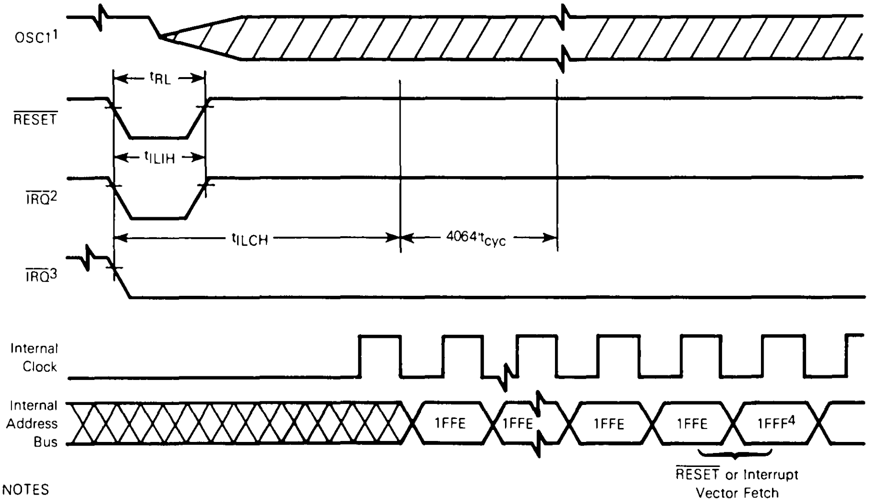
(V_{DD}=5.0 Vdc±10%, V_{SS}=0 Vdc, T_A=T_L to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	8.0	
External Clock Option		dc	8.0	
Internal Operating Frequency	f_{op}			MHz
Crystal (f_{osc}÷2)		—	4.0	
External Clock (f_{osc}−2)		dc	4.0	
Cycle Time (see Figure 28)	t_{cyc}	250	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 26)	t_{TH}, t_{TL}	63	—	ns
Input Capture Pulse Period (see Figure 26)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	63	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	45	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{IRQ}$ pin edge-sensitive mask option
3 $\overline{IRQ}$ pin level and edge-sensitive mask option
4 $\overline{RESET}$ vector address shown for timing example

Figure 25. Stop Recovery Timing Diagram

CONTROL TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 2.0 2.0	MHz
Internal Operating Frequency Crystal ($f_{osc} - 2$) External Clock ($f_{osc} - 2$)	f_{op}	 — dc	 1.0 1.0	MHz
Cycle Time (see Figure 28)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 26) Input Capture Pulse Period (see Figure 26)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

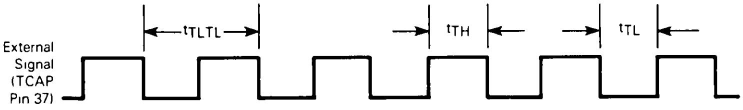


Figure 26. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	$f_{op(s)}$
	Slave	$f_{op(s)}$	dc	4.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	250	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	TBD	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	TBD	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	TBD	—	ns
	Slave	$t_{w(SCKH)s}$	TBD	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	TBD	—	ns
	Slave	$t_{w(SCKL)s}$	TBD	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	TBD	—	ns
	Slave	$t_{su(s)}$	TBD	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	TBD	—	ns
	Slave	$t_{h(s)}$	TBD	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	TBD	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	TBD	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	TBD	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, $C_L=200$ pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	TBD	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, $C_L=200$ pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	TBD	µs

*Signal production depends on software.

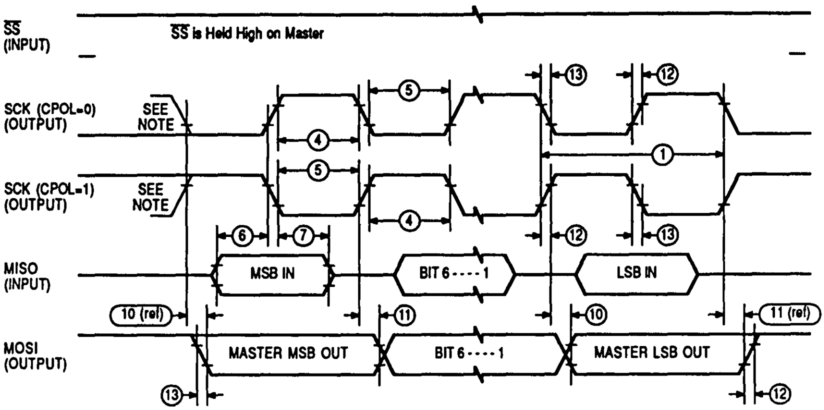
**Assumes 200 pF load on all SPI pins.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

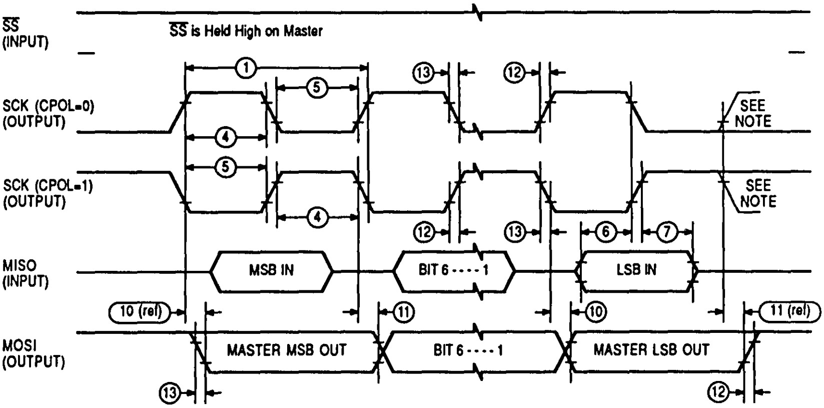
Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 1.0	 — —	 t_{cyc} µs
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 500	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 500	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 720 400	 — —	 µs ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 720 400	 — —	 µs ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 200 200	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 200 200	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 250	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 500	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 500	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 200 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 200 2.0	 ns µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6----1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6----1
MASTER LSB OUT
10 (ref)
11 (ref)
1
4
5
6
7
10
11
12
13

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

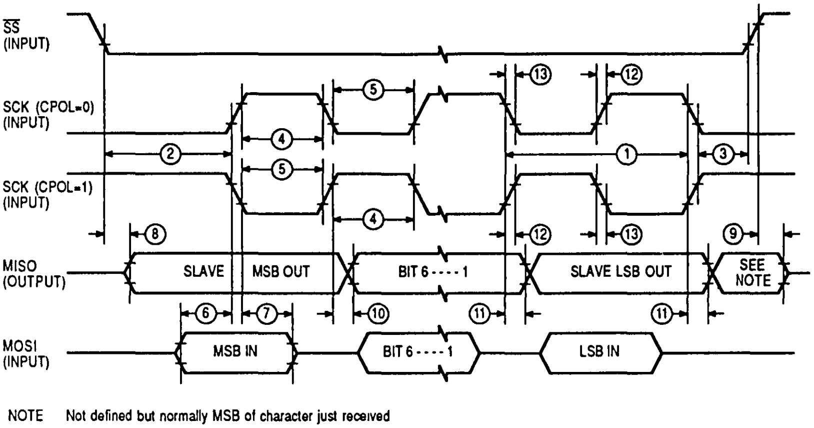
a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6----1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6----1
MASTER LSB OUT
10 (ref)
11 (ref)
1
4
5
6
7
10
11
12
13

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

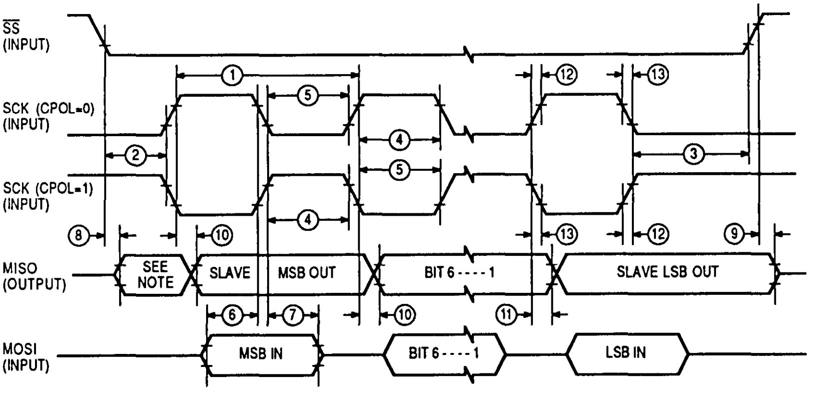
b) SPI MASTER TIMING (CPHA=1)

Figure 27. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally MSB of character just received

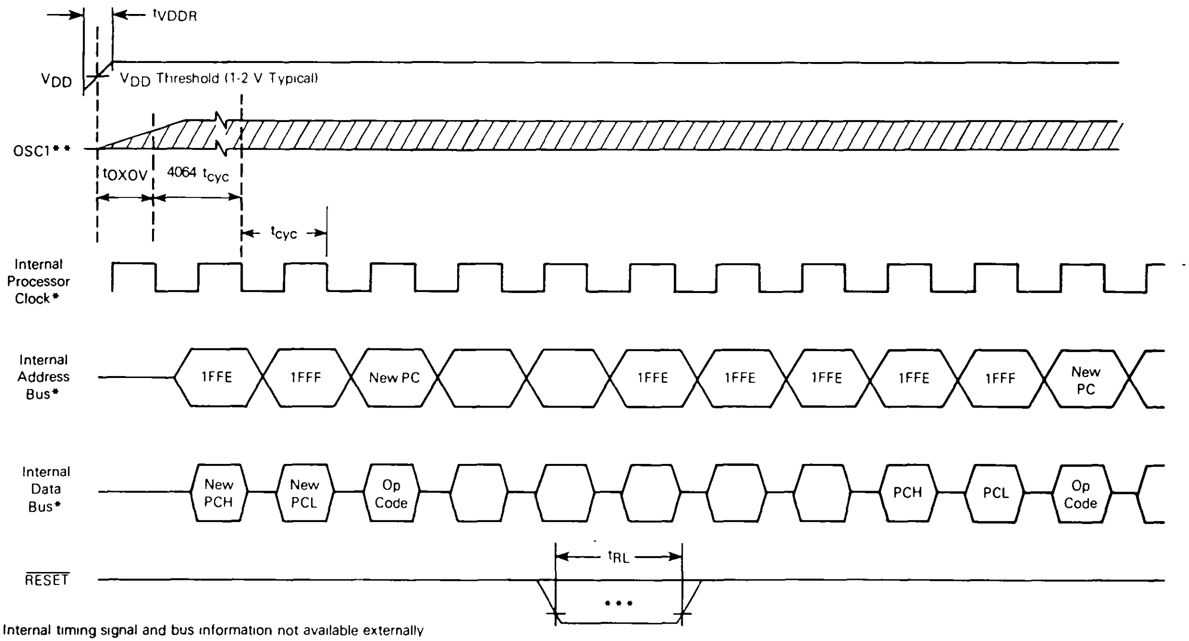
c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE: Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA=1)

Figure 27. SPI Timing Diagrams (Sheet 2 of 2)


tVDDR
VDD
VDD Threshold (1-2 V Typical)
OSC1**
tOXOV
4064 tcyc
tcyc
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
RESET
tRL
***

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 28. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS™, disk file
MS™-DOS/PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

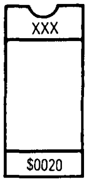
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2764, 68764, or 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

Start the page zero, user ROM at EEPROM address \$0020 through \$004F. Start the user ROM at EEPROM address \$0100 through \$10EF with vectors from \$1FF4 to \$1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

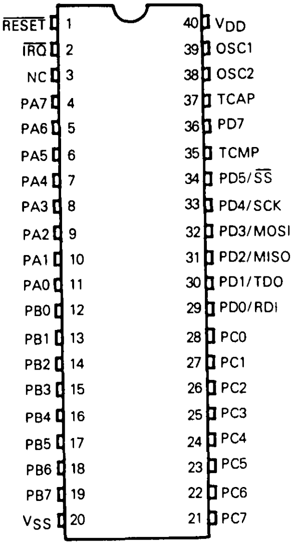
The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05C4 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C	MC68HSC05C4P
PLCC (FN Suffix)	0°C to +70°C	MC68HSC05C4FN

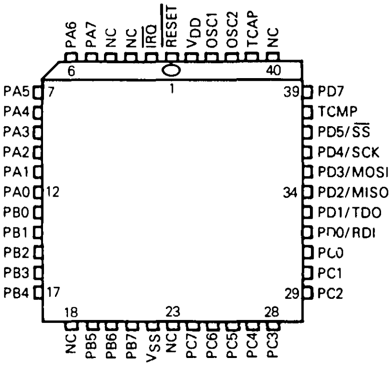
MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE


RESET 1
IRQ 2
NC 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE


PA6
PA7
NC
NC
IRQ
RESET
VDD
OSC1
OSC2
TCAP
NC
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
VSS
NC
PC7
PC6
PC5
PC4
PC3

NOTE: Bulk substrate tied to V_{SS}.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HSC05C8

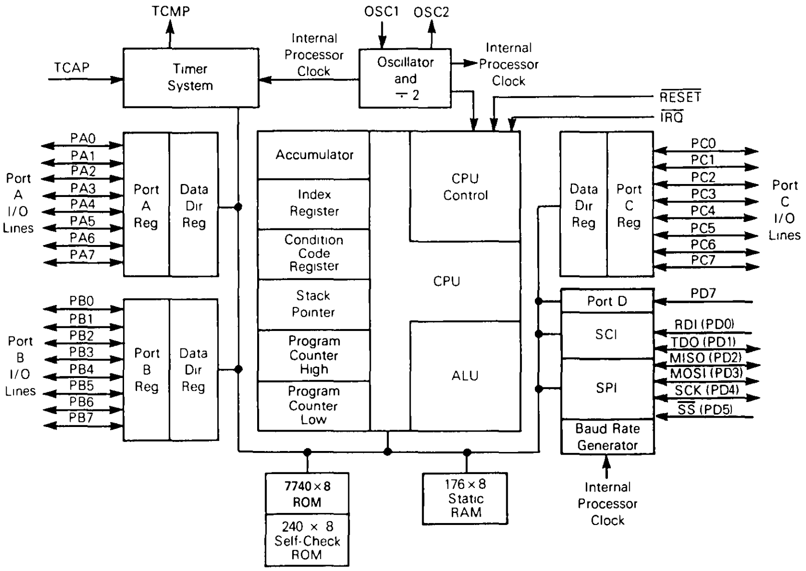
Technical Summary
8-Bit Microcontroller Unit

The MC68HSC05C8 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 7740 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8 × 8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
TCAP
Timer System
Internal Processor Clock
OSC1
OSC2
Oscillator and ÷ 2
Internal Processor Clock
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I/O Lines
Port A Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
SS (PD5)
Baud Rate Generator
Internal Processor Clock
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I/O Lines
Port B Reg
Data Dir Reg
7740 × 8 ROM
240 × 8 Self-Check ROM
176 × 8 Static RAM

This document contains information on a new product Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

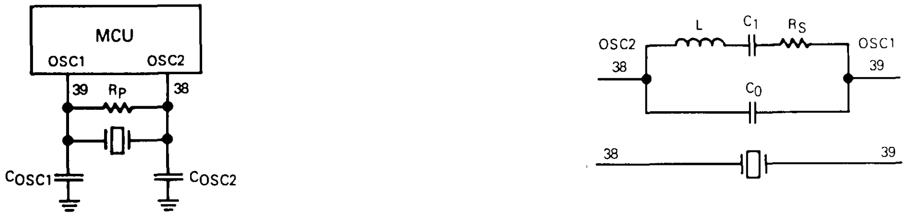
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

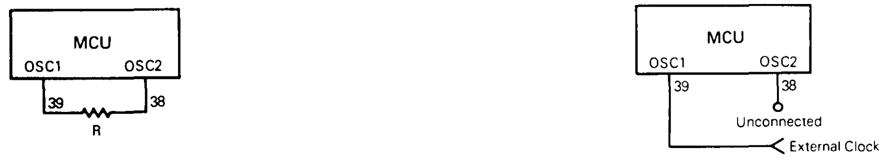
	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

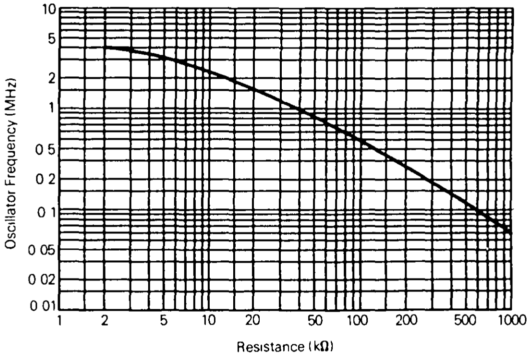


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{\text{W}}$ is an internal signal.

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

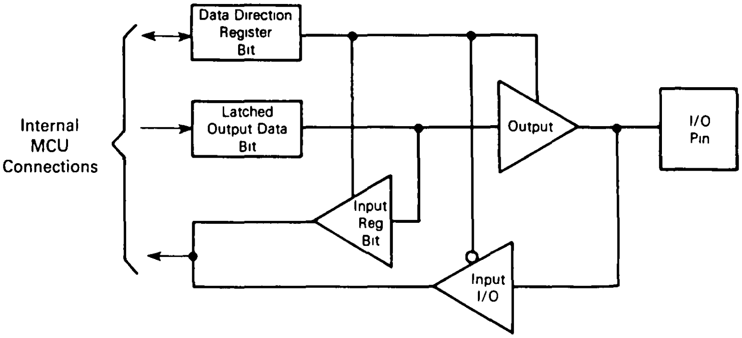


Figure 3. Typical Port I/O Circuit

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

7	0
A	

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.

7	0
X	

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.

12	0
PC	

STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

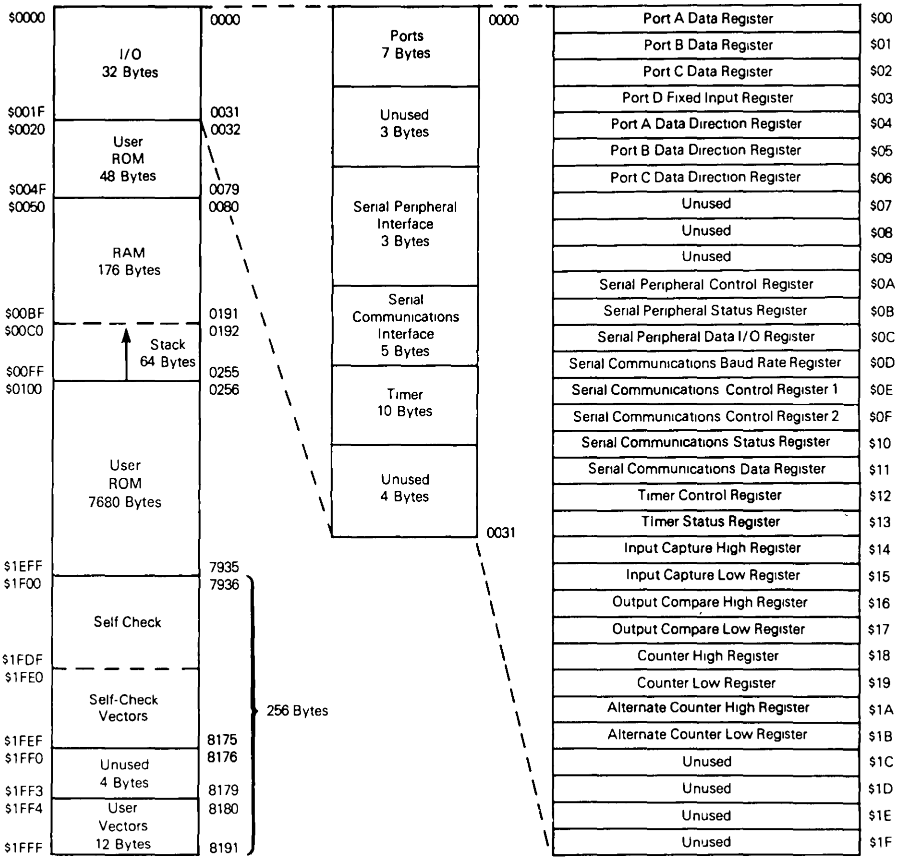
When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

12						7	0
0	0	0	0	0	1	1	SP

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

CCR				
H	I	N	Z	C



$0000
I/O
32 Bytes
$001F
$0020
User
ROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
User
ROM
7680 Bytes
$1EFF
$1F00
Self Check
$1FDF
$1FE0
Self-Check
Vectors
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
7935
7936
8175
8176
8179
8180
8191
256 Bytes
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Unused
4 Bytes
0000
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Unused $1C
Unused $1D
Unused $1E
Unused $1F

Figure 4. Memory Map

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

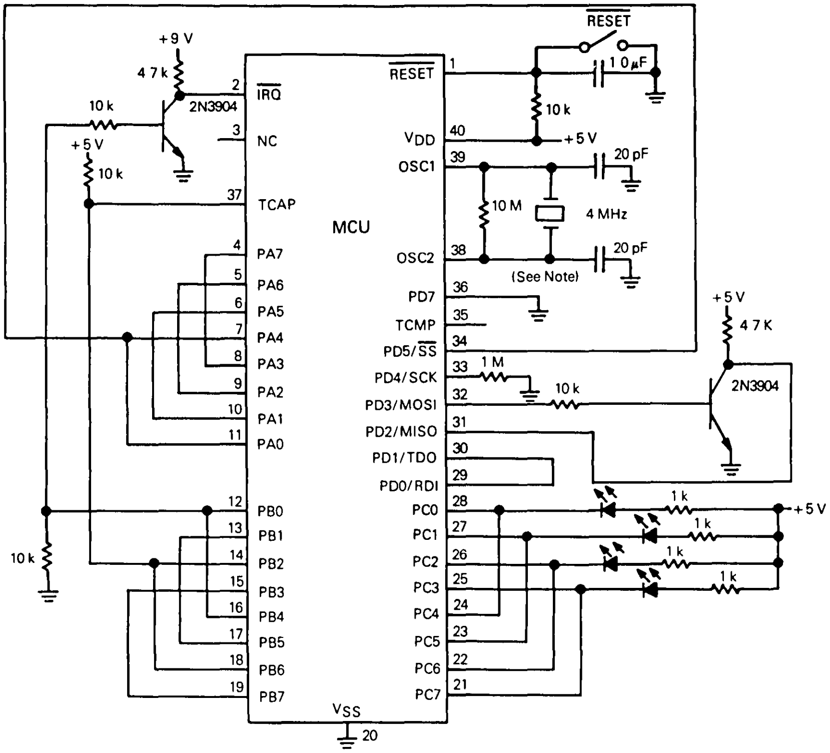
Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF flag


RESET
+9 V
4.7 k
2
IRQ
2N3904
10 k
3
NC
+5 V
10 k
37
TCAP
MCU
4
PA7
5
PA6
6
PA5
7
PA4
8
PA3
9
PA2
10
PA1
11
PA0
12
PB0
13
PB1
14
PB2
15
PB3
16
PB4
17
PB5
18
PB6
19
PB7
10 k
VSS
20
RESET
1
1.0 μF
10 k
VDD
40
+5 V
OSC1
39
20 pF
10 M
4 MHz
OSC2
38
20 pF
(See Note)
PD7
36
TCMP
35
PD5/SS
34
PD4/SCK
33
1 M
PD3/MOSI
32
10 k
PD2/MISO
31
PD1/TDO
30
PD0/RDI
29
+5 V
4.7 K
2N3904
PC0
28
PC1
27
PC2
26
PC3
25
PC4
24
PC5
23
PC6
22
PC7
21
1 k
1 k
1 k
1 k
+5 V

NOTE The RC Oscillator Option may also be used in this circuit

Figure 5. Self-Check Circuit Schematic Diagram

Interrupts — Tests external, timer, SCI and SPI interrupts

SCI — Transmission test; checks RDRF, TDRE, TC, and FE flags

SPI — Transmission test; checks SPIF, WCOL, and MODF flags

Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or IRQ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on; 1 indicates LED is off.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM checksum subroutine is called at location $1F93 with RAM location $0053 equal to $01 and A=0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations $0050 through $0053 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input (RESET). The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the RESET pin is low at the end of 4064 t_{cyc}, the MCU will remain in the reset condition until RESET goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts (IRQ, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

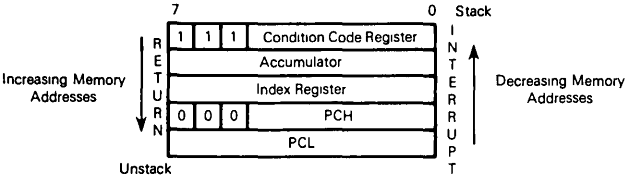
EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at IRQ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.


7
0
Stack
RETURN
1 1 1 Condition Code Register
Accumulator
Index Register
0 0 0 PCH
PCL
Unstack
INTERRUPT
Increasing Memory Addresses
Decreasing Memory Addresses

NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 6. Interrupt Stacking Order

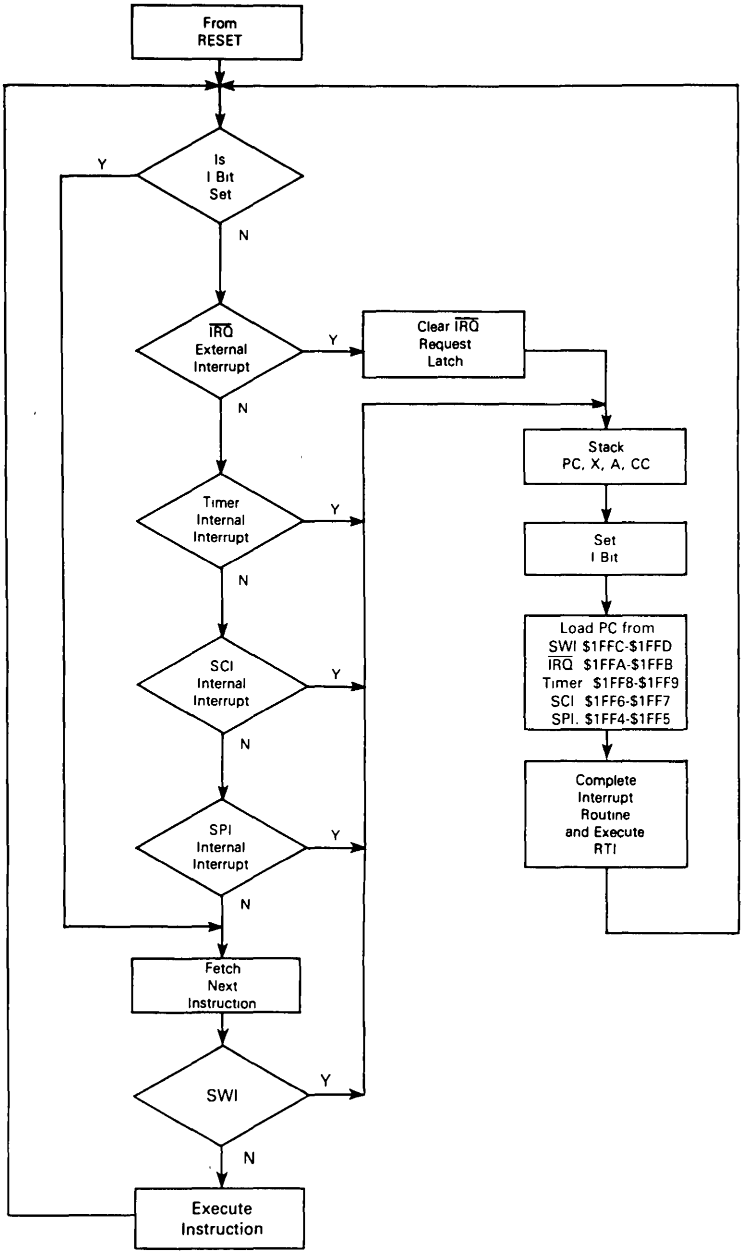
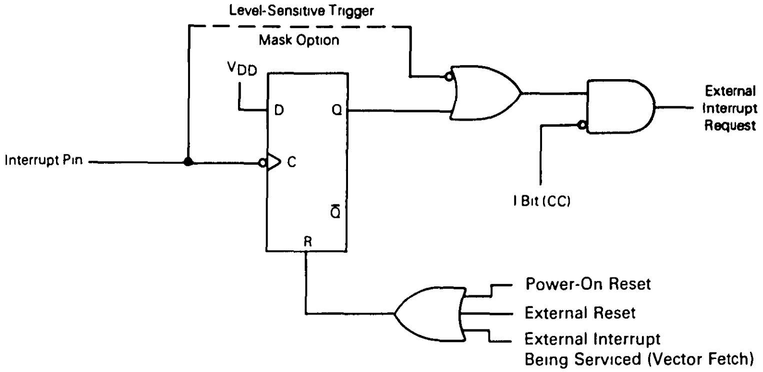
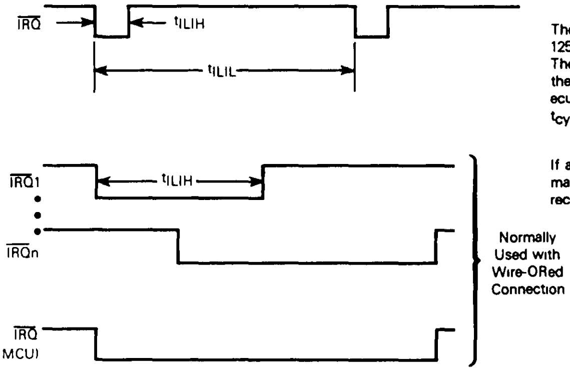

From RESET
Is I Bit Set
Y
N
$\overline{\text{IRQ}}$ External Interrupt
Y
Clear $\overline{\text{IRQ}}$ Request Latch
N
Stack PC, X, A, CC
Timer Internal Interrupt
Y
N
Set I Bit
Load PC from
SWI $1FFC-$1FFD
$\overline{\text{IRQ}}$ $1FFA-$1FFB
Timer $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI. $1FF4-$1FF5
SCI Internal Interrupt
Y
N
SPI Internal Interrupt
Y
Complete Interrupt Routine and Execute RTI
N
Fetch Next Instruction
SWI
Y
N
Execute Instruction

Figure 7. Reset and Interrupt Processing Flowchart


Level-Sensitive Trigger
Mask Option
V_{DD}
D
Q
C
$\overline{Q}$
R
Interrupt Pin
External Interrupt Request
I Bit (CC)
Power-On Reset
External Reset
External Interrupt Being Serviced (Vector Fetch)

(a) Interrupt Internal Function Diagram


$\overline{IRQ}$
t_{ILIH}
t_{ILIL}
$\overline{IRQ}1$
$\overline{IRQ}n$
$\overline{IRQ}$ (MCU)
Normally Used with Wire-ORed Connection

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

(b) Interrupt Mode Diagram

Figure 8. External Interrupt

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 9).

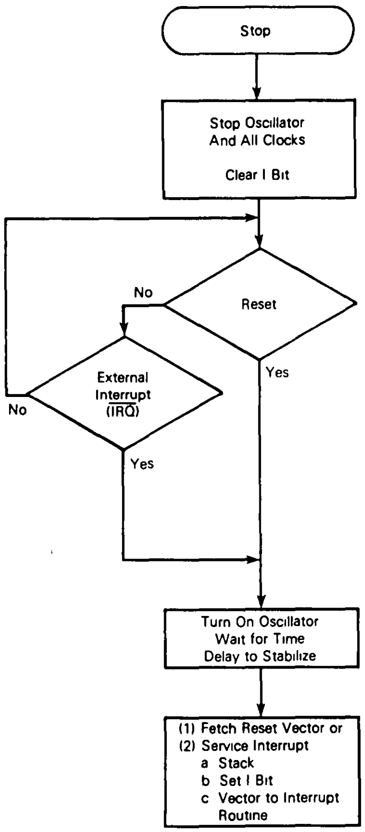


Figure 9. STOP Function Flowchart

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI during STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the IRQ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the IRQ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the IRQ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active (refer to Figure 10). An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in RESET during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

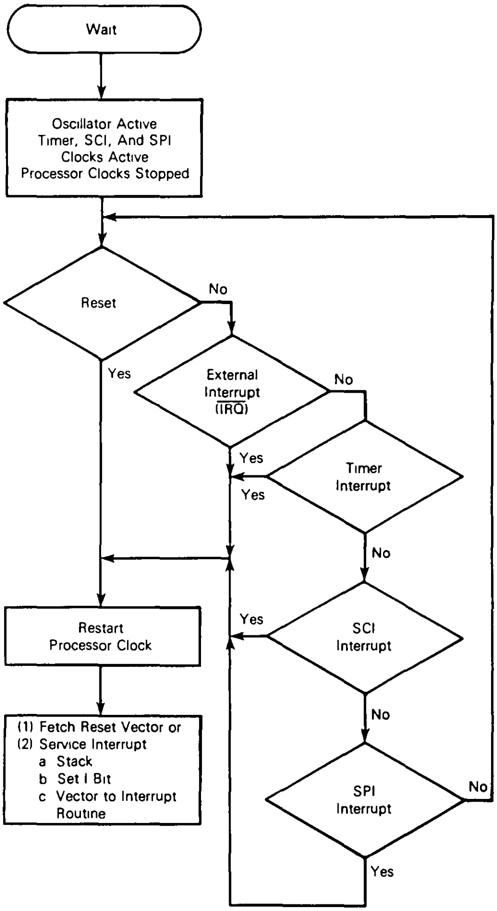

Wait
Oscillator Active
Timer, SCI, And SPI
Clocks Active
Processor Clocks Stopped
Reset
No
Yes
External
Interrupt
(IRQ)
No
Yes
Timer
Interrupt
Yes
No
Restart
Processor Clock
Yes
SCI
Interrupt
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt
Routine
SPI
Interrupt
No
Yes

Figure 10. WAIT Function Flowchart

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18–$19 (counter register) or $1A–$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins

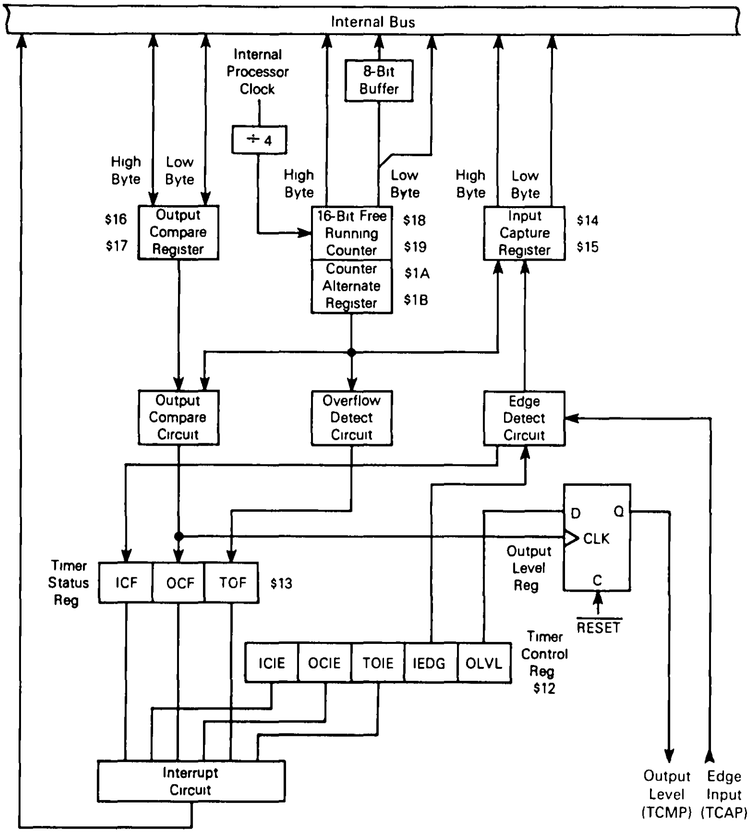

Internal Bus
Internal Processor Clock
8-Bit Buffer
÷ 4
High Byte
Low Byte
High Byte
Low Byte
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
$18
$19
Counter Alternate Register
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
Timer Control Reg $12
ICIE
OCIE
TOIE
IEDG
OLVL
Interrupt Circuit
Output Level (TCMP)
Edge Input (TCAP)

Figure 11. Timer Block Diagram

running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

IEDG — Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG bit (U = unaffected).

OLVL — Output Level
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
1 = High output
0 = Low output

Bits 2, 3, and 4 — Not used
Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF — Output Compare Flag
1 = Flag set when output compare register contents match the free-running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 12.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 13); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register

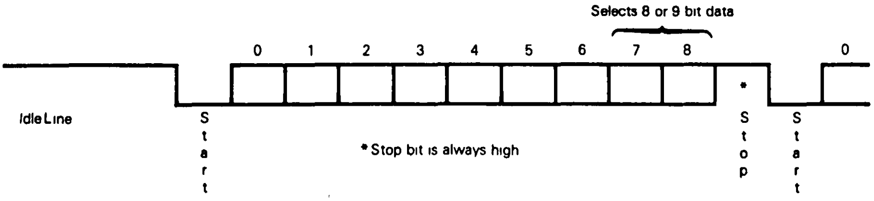


Figure 12. Data Format

is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 13. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates, which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET:	U	U	U	U	U	U	U	U

As shown in Figure 13, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $OE

The SCCR1 provides control bits that determine word length and select the wake-up method.

	7	6	5	4	3	2	1	0
	R8	T8	—	M	WAKE	—	—	—
RESET	U	U	—	U	U	—	—	—

R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

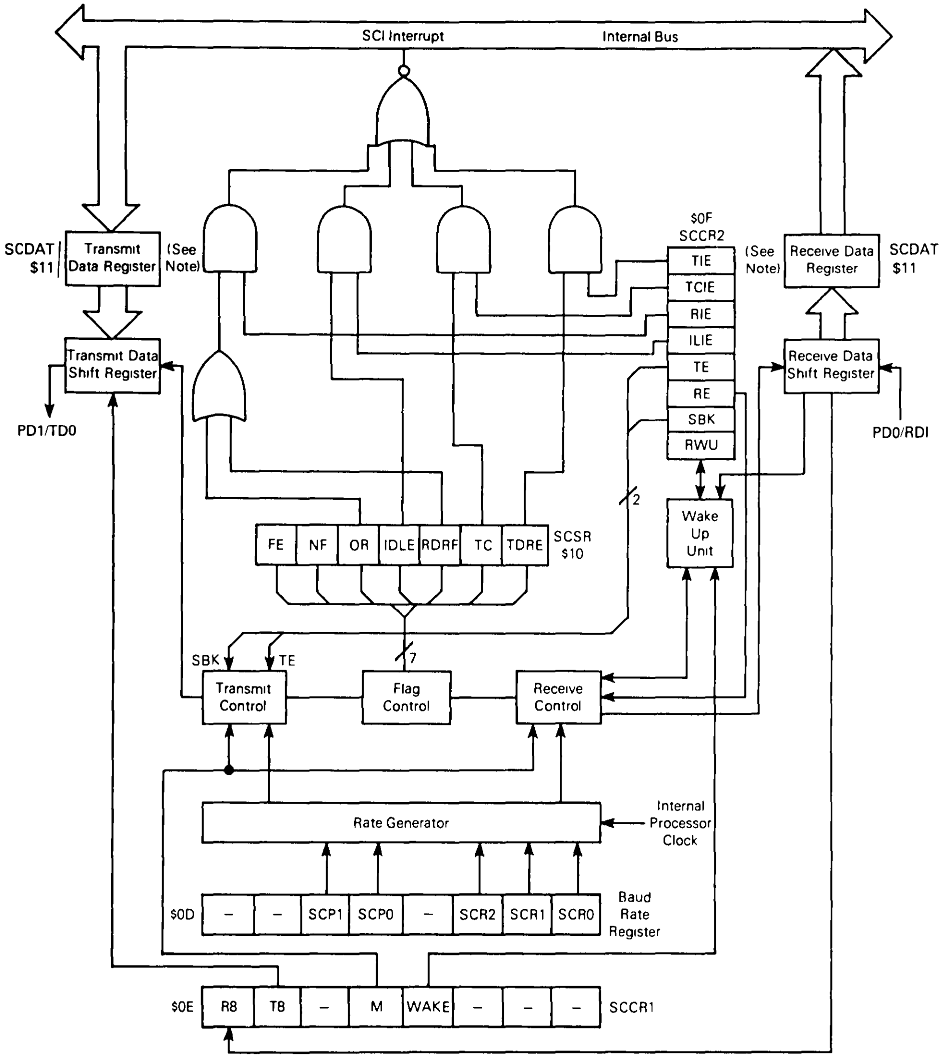
M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most significant bit)
0 = Idle line condition

Bits 0–2, and 5 — Not used
Can read either one or zero

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.


SCI Interrupt
Internal Bus
SCDAT
$11
Transmit Data Register
(See Note)
Transmit Data Shift Register
PD1/TDO
$0F
SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(See Note)
Receive Data Register
SCDAT
$11
Receive Data Shift Register
PD0/RDI
2
Wake Up Unit
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR
$10
7
SBK
TE
Transmit Control
Flag Control
Receive Control
Rate Generator
Internal Processor Clock
$0D
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
Baud Rate Register
$0E
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 13. SCI Block Diagram

Serial Communications Control Register 2 (SCCR2) $OF

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET.								
	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt enabled
0 = TDRE interrupt disabled

TCIE — Transmit Complete Interrupt Enable
1 = SCI interrupt enabled
0 = TC interrupt disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt enabled
0 = RDRF and OR interrupts disabled

ILIE — Idle Line Interrut Enable
1 = SCI interrupt enabled
0 = Idle interrupt disabled

TE — Transmit Enable
1 = Transmit shift register output is applied to the TDO line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TDO line becomes a high-impedance line.

RE — Receive Enable
1 = Receiver shift register input is applied to the RDI line.
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.

RWU — Receiver Wake-Up
1 = Places receiver in sleep mode and enables wake-up function
0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)

SBK — Send Break
1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET	1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
1 = TDR contents transferred to the transmit data shift register
0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1), followed by a write to the TDR.

TC — Transmit Complete
1 = Indicates end of data frame, preamble, or break condition has occurred
0 = TC bit cleared by reading the SCSR (with TC = 1), followed by a write to the TDR

RDRF — Receive Data Register (RDR) Full
1 = Receive data shift register contents transferred to the RDR
0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR

IDLE — Idle Line Detect
1 = Indicates receiver has detected an idle line
0 = IDLE is cleared by reading the SCSR (with IDLE = 1), followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error
1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
0 = OR is cleared by reading the SCSR (with OR = 1), followed by a read of the RDR.

NF — Noise Flag
1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
0 = NF is cleared by reading the SCSR (with NF = 1), followed by a read of the RDR.

FE — Framing Error
1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.
0 = NF is cleared by reading the SCSR (with FE = 1), followed by a read of the RDR.

Bit 0 — Not used
Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

	7	6	5	4	3	2	1	0
	—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET	—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0
SCP1 — SCI Prescaler Bit 1
Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 3.

SCR0 — SCI Baud Rate Bit 0
SCR1 — SCI Baud Rate Bit 1
SCR2 — SCI Baud Rate Bit 2
Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 4.

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz					
1	0		8.0	4.194304	4.0	2.4576	2.0	1.8432
0	0	1	250.000 kHz	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	83.332 kHz	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	62.500 kHz	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	19.200 kHz	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output					
2	1	0		250.000 kHz	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	—	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	125.000 kHz	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	62.500 kHz	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	31.250 kHz	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	15.625 kHz	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	7.813 kHz	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	3.906 kHz	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.953 kHz	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided-by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600-Hz baud rate is required with a 2.4576-MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 14) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 2.0 MHz (maximum) master bit frequency
- 4.0 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}=1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The

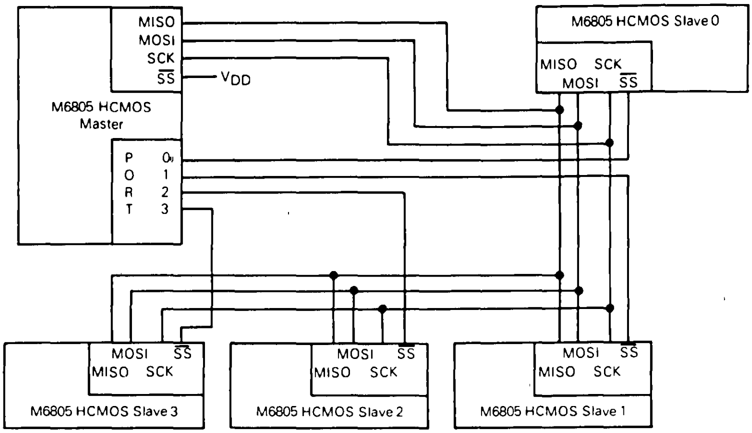

MISO
MOSI
SCK
$\overline{SS}$
V_{DD}
M6805 HCMOS Master
P O R T
0
1
2
3
M6805 HCMOS Slave 0
MISO SCK
MOSI $\overline{SS}$
MOSI $\overline{SS}$
MISO SCK
M6805 HCMOS Slave 3
M6805 HCMOS Slave 2
M6805 HCMOS Slave 1

Figure 14. Master-Slave System Configuration

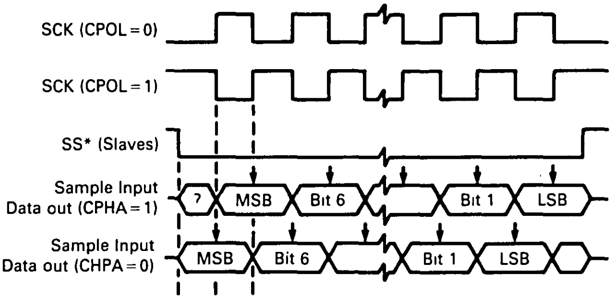

SCK (CPOL=0)
SCK (CPOL=1)
SS* (Slaves)
Sample Input
Data out (CPHA=1)
?
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA=0)
MSB
Bit 6
Bit 1
LSB

Figure 15. Data Clock Timing Diagram

master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 15, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{SS}$) input line selects a slave device. The $\overline{SS}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high; if the $\overline{SS}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{SS}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 16. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

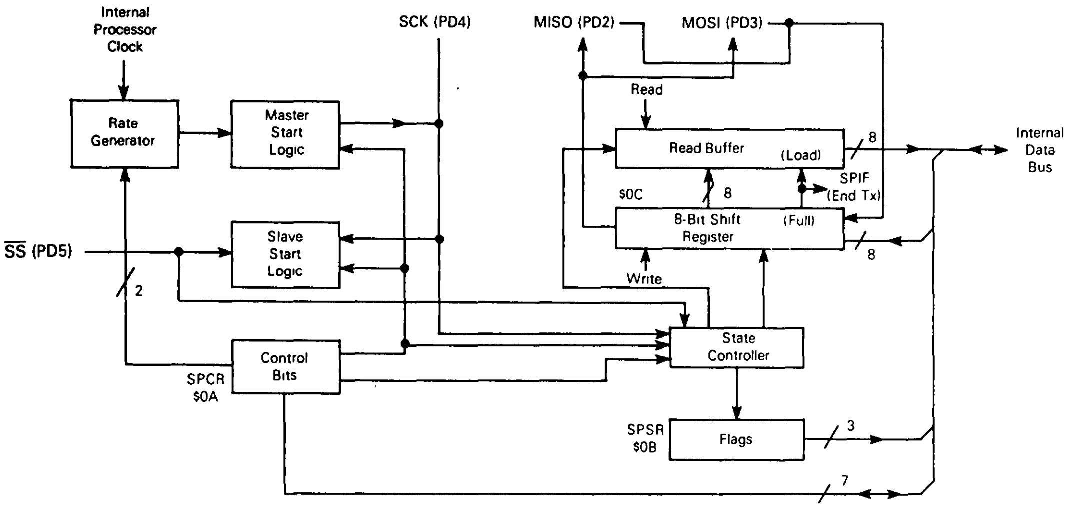

Internal Processor Clock
Rate Generator
Master Start Logic
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Read
Read Buffer
(Load)
8
Internal Data Bus
SPIF (End Tx)
$0C
8
8-Bit Shift Register
(Full)
8
SS (PD5)
Slave Start Logic
2
Write
State Controller
Control Bits
SPCR $0A
SPSR $0B
Flags
3
7

Figure 16. SPI Block Diagram

In a slave configuration, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 17 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

	7	6	5	4	3	2	1	0
	SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:	0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
- 1 = SPI interrupt enabled
- 0 = SPI interrupt disabled

SPE — Serial Peripheral System Enable
- 1 = SPI system on
- 0 = SPI system off

MSTR — Master Mode Select
- 1 = Master mode
- 0 = Slave mode

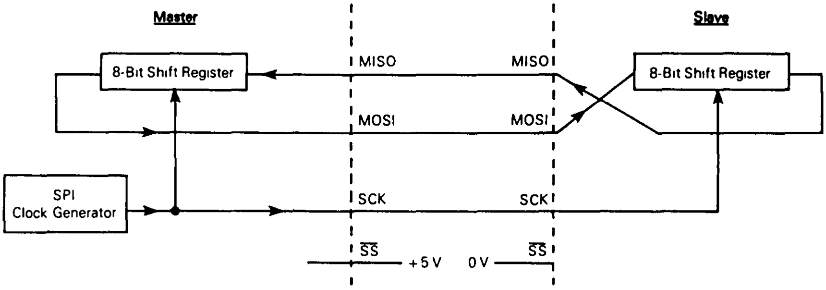

Master
Slave
8-Bit Shift Register
MISO
MISO
MOSI
MOSI
8-Bit Shift Register
SPI Clock Generator
SCK
SCK
SS
+5 V
0 V
SS

Figure 17. SPI Master-Slave Interconnections

CPOL — Clock Polarity

Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.

1 = SCK line idles high

0 = SCK line idles in low state

CPHA — Clock Phase

Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.

1 = $\overline{SS}$ is an output enable control.

0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits

Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 – Not used

Can read either one or zero

SPI Clock Rate Selection

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—
RESET.							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag

1 = Indicates data transfer completed between processor and external device.
(If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)

0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.

WCOL — Write Collision

1 = Indicates an attempt is made to write to SPDR while data transfer is in process.

0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.

MODF — Mode Fault Flag

1 = Indicates multi-master system control conflict.

0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 0–3, and 5 — Not used

Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read/write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

<table>
<tr><td>Operation</td><td colspan="4">X:A X*A</td></tr>
<tr><td>Description</td><td colspan="4">Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register</td></tr>
<tr><td>Condition Codes</td><td colspan="4">H: Cleared
I: Not affected
N: Not affected
Z: Not affected
C: Cleared</td></tr>
<tr><td>Source</td><td colspan="4">MUL</td></tr>
<tr><td rowspan="2">Form(s)</td><td>Addressing Mode</td><td>Cycles</td><td>Bytes</td><td>Opcode</td></tr>
<tr><td>Inherent</td><td>11</td><td>1</td><td>$42</td></tr>
</table>

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX

— Continued —

Function	Mnemonic
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI

— Continued —

Function	Mnemonic
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is

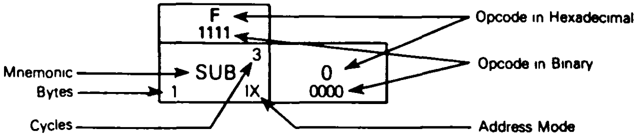
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

[illegible]

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS

($V_{DD} = 5.0$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage ($I_{Load} = 0.8$ mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) ($I_{Load} = 1.6$ mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 21) ($I_{Load} = 1.6$ mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 23) Wait (see Figures 22 and 23) Stop (see Figure 23) 25°C 0° to 70°C (Standard)	I_{DD}	 — — — —	 6.7 3.0 2.0 —	 13.3 7.6 50 140	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source ($f_{osc} = 8.0$ MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, $C_L = 20$ pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, $V_{IL} = 0.2$ V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.3$ $V_{DD} - 0.3$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 24) Wait (see Figures 22 and 24) Stop (see Figure 24) 25°C 0° to 70°C (Standard)	I_{DD}	 — — — —	 1.0 0.5 1.0 —	 2.5 1.4 30 80	 mA mA μA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0) If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. A 25°C only version is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

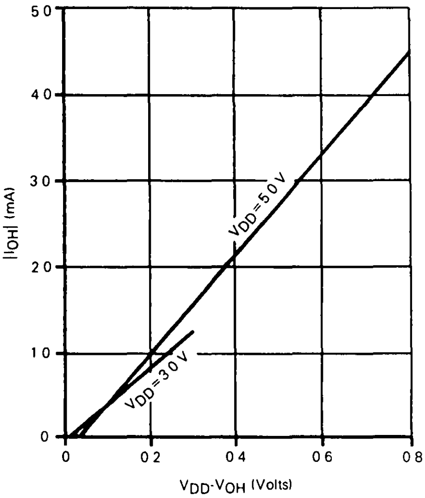


Figure 20. Typical V_{OH} vs I_{OH} for PD1-PD4

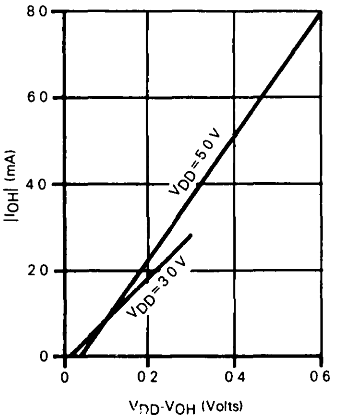


Figure 19. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

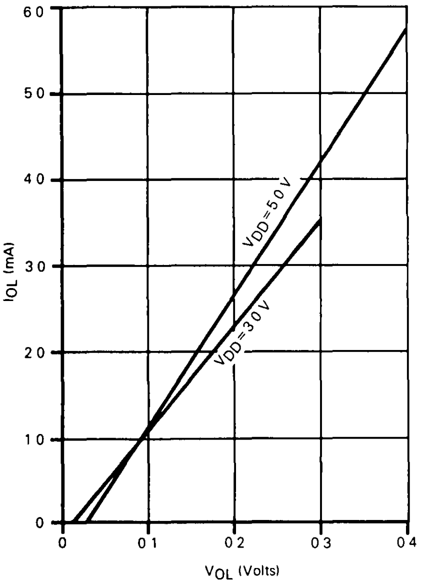


Figure 21. Typical V_{OL} vs I_{OL} for All Ports

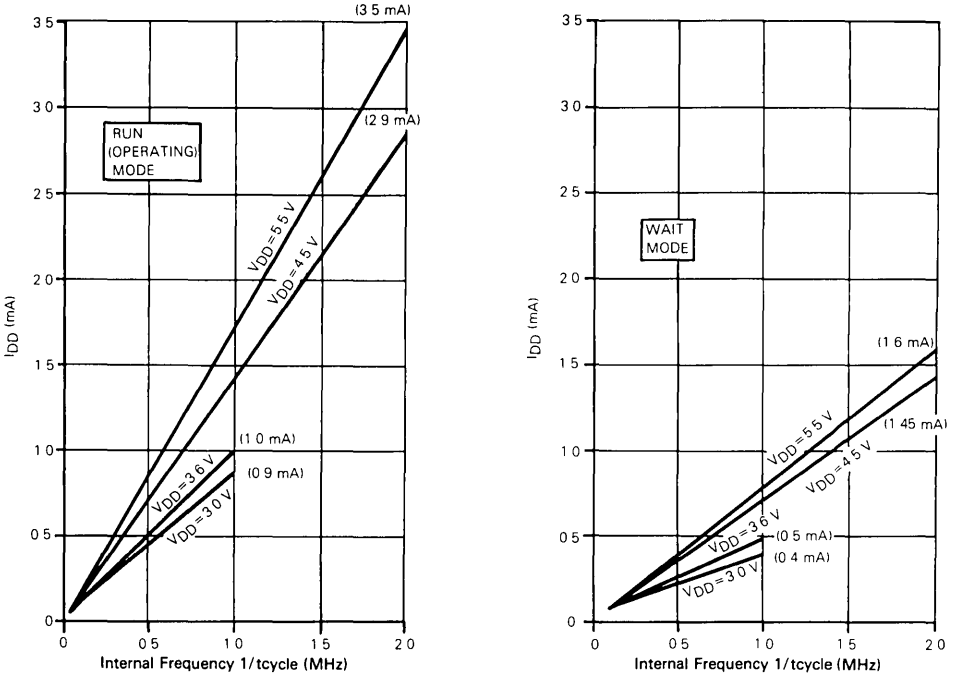


Figure 22. Typical Current vs Internal Frequency for Run and Wait Modes

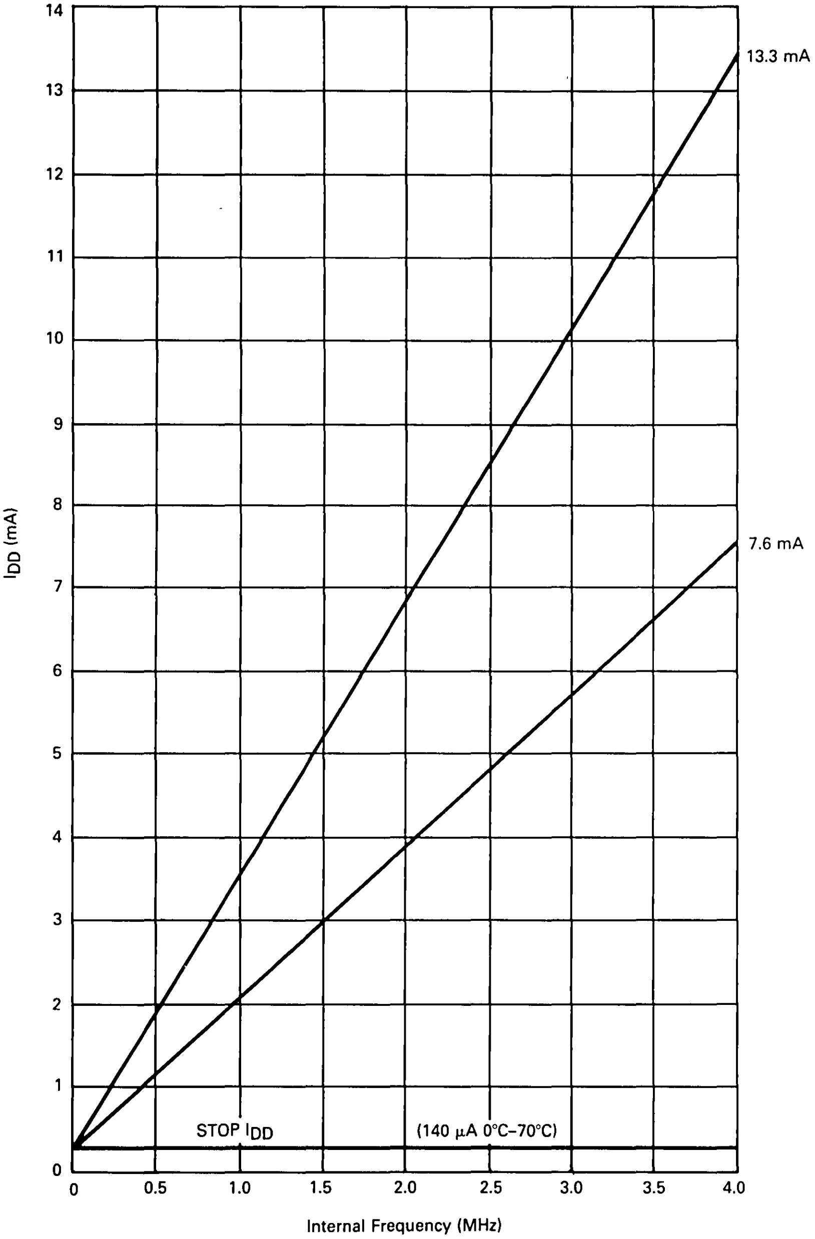


Figure 23. Maximum I_{DD} vs Frequency for $V_{DD}=5.0$ Vdc

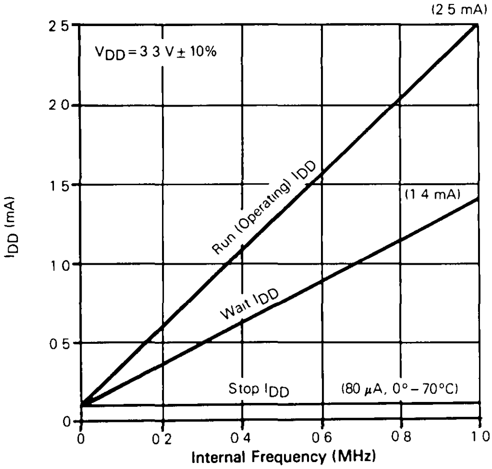


Figure 24. Maximum I_{DD} vs Frequency for V_{DD} = 3.3 Vdc

CONTROL TIMING

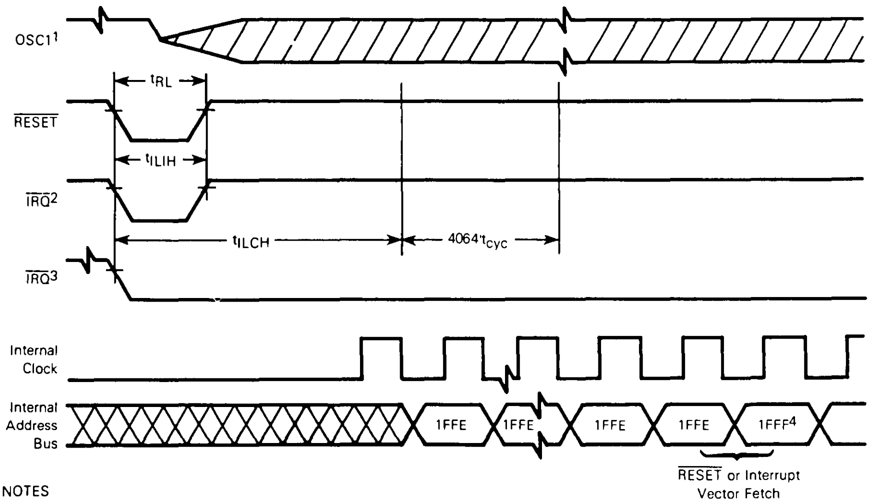
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	8.0	
External Clock Option		dc	8.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} \div 2$)		—	4.0	
External Clock ($f_{osc} - 2$)		dc	4.0	
Cycle Time (see Figure 28)	t_{cyc}	250	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 26)	t_{TH}, t_{TL}	63	—	ns
Input Capture Pulse Period (see Figure 26)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	63	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	45	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3 $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option
4 $\overline{\text{RESET}}$ vector address shown for timing example

Figure 25. Stop Recovery Timing Diagram

CONTROL TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} \div 2$)		—	1.0	
External Clock ($f_{osc} \div 2$)		dc	1.0	
Cycle Time (see Figure 28)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 26)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 26)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

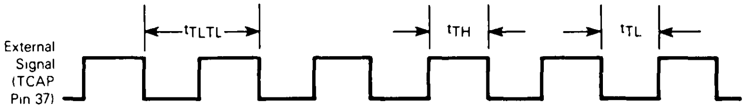


Figure 26. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0,5	$f_{op(s)}$
	Slave	$f_{op(s)}$	dc	4.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	250	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	TBD	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	TBD	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	TBD	—	ns
	Slave	$t_{w(SCKH)s}$	TBD	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	TBD	—	ns
	Slave	$t_{w(SCKL)s}$	TBD	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	TBD	—	ns
	Slave	$t_{su(s)}$	TBD	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	TBD	—	ns
	Slave	$t_{h(s)}$	TBD	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	TBD	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	TBD	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	TBD	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	TBD	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	TBD	µs

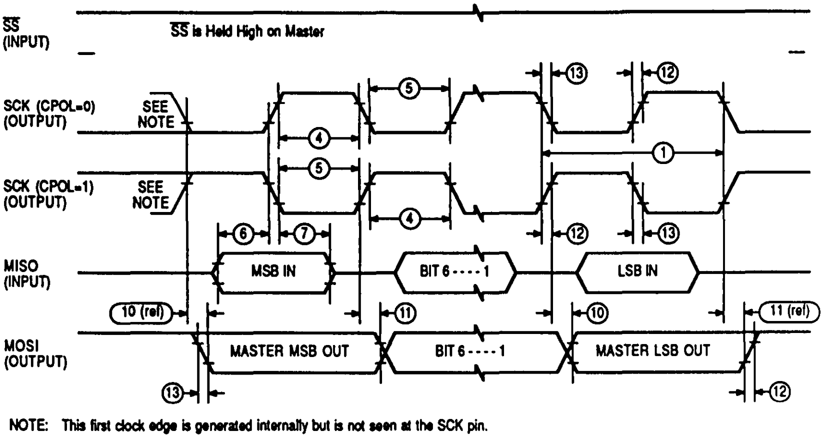
*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING
(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

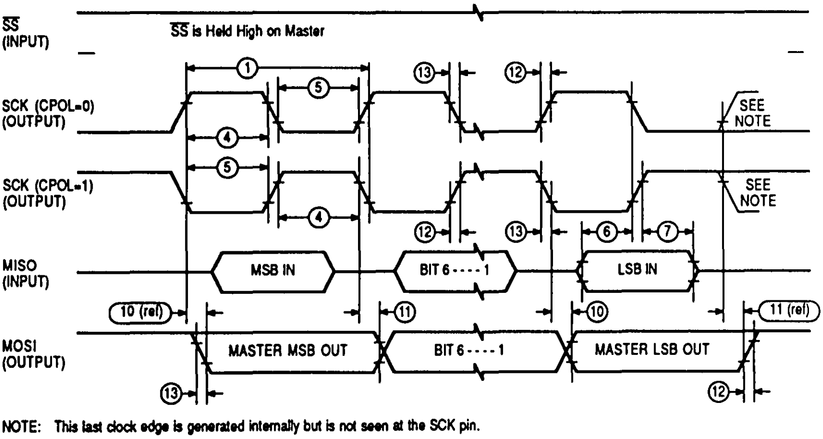
Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	$f_{op(s)}$
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	µs
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	720	—	µs
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	720	—	µs
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	250	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	500	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

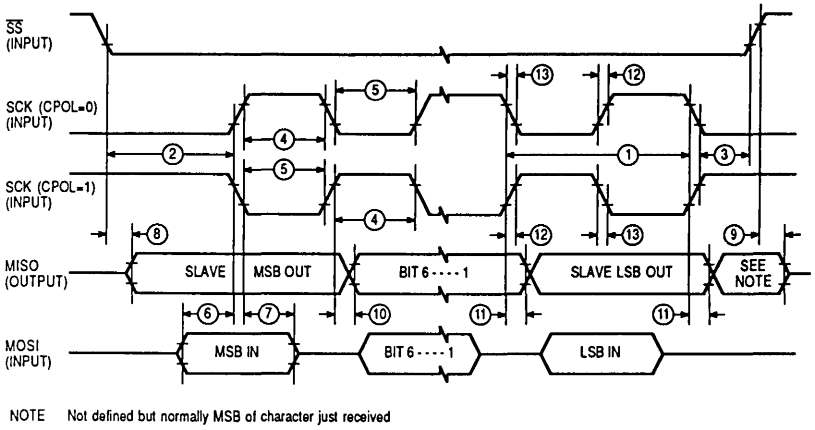
a) SPI MASTER TIMING (CPHA = 0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

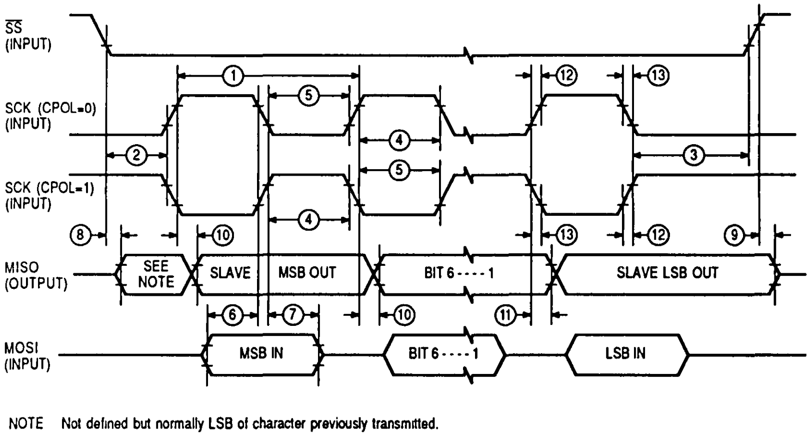
b) SPI MASTER TIMING (CPHA = 1)

Figure 27. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally MSB of character just received

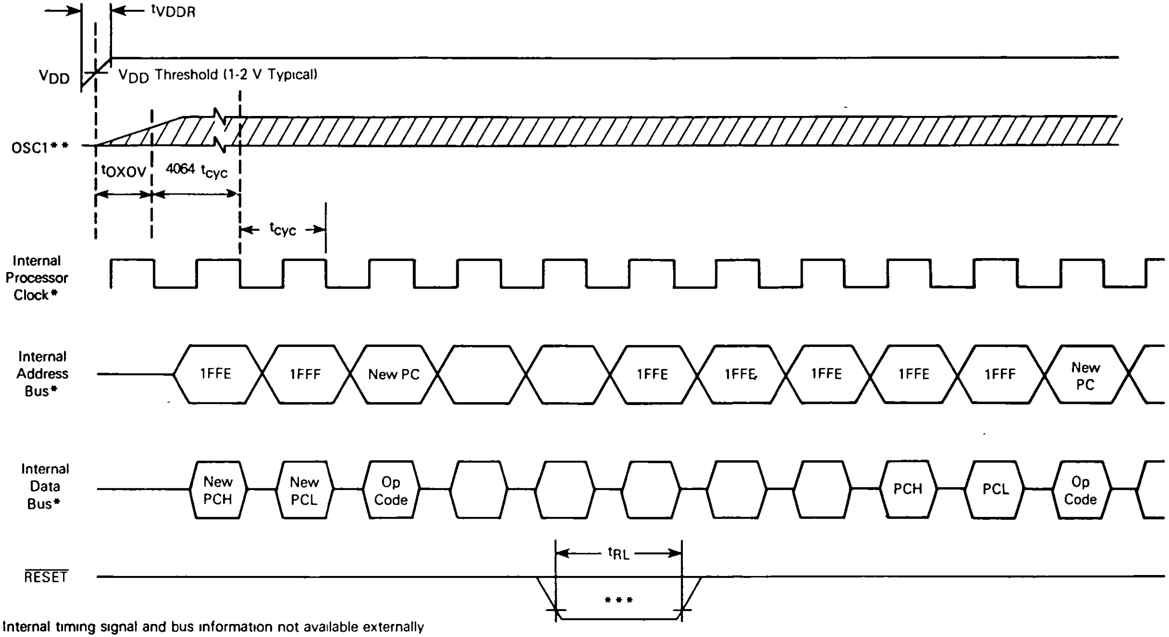
c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA=1)

Figure 27. SPI Timing Diagrams (Sheet 2 of 2)


tVDDR
VDD
VDD Threshold (1-2 V Typical)
OSC1**
tOXOV
4064 tcyc
tcyc
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
tRL
RESET
***

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency. It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 28. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS™, disk file
MS™-DOS/PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

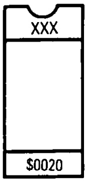
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2764, 68764, or 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

Start the page zero, user ROM at EEPROM address \$0020 through \$004F. Start the user ROM at EEPROM address \$0100 through \$1EFF with vectors from \$1FF4 to \$1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

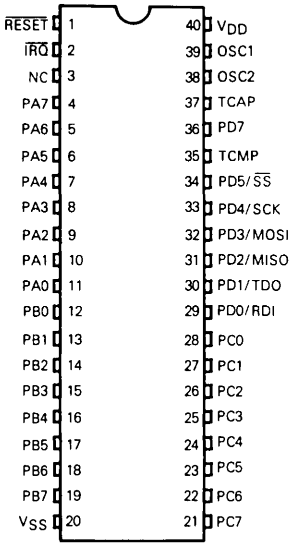
The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05C4 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C	MC68HSC05C8P
PLCC (FN Suffix)	0°C to +70°C	MC68HSC05C8FN

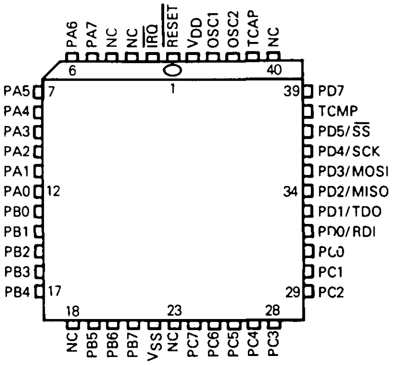
MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE


RESET 1
IRQ 2
NC 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE


PA6
PA7
NC
NC
IRQ
RESET
VDD
OSC1
OSC2
TCAP
NC
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
VSS
NC
PC7
PC6
PC5
PC4
PC3

NOTE: Bulk substrate tied to V_{SS}.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC705B5

Product Preview

One Time Programmable ROM (OTPROM) or Standard Eraseable Programmable ROM (EPROM) Microcontroller

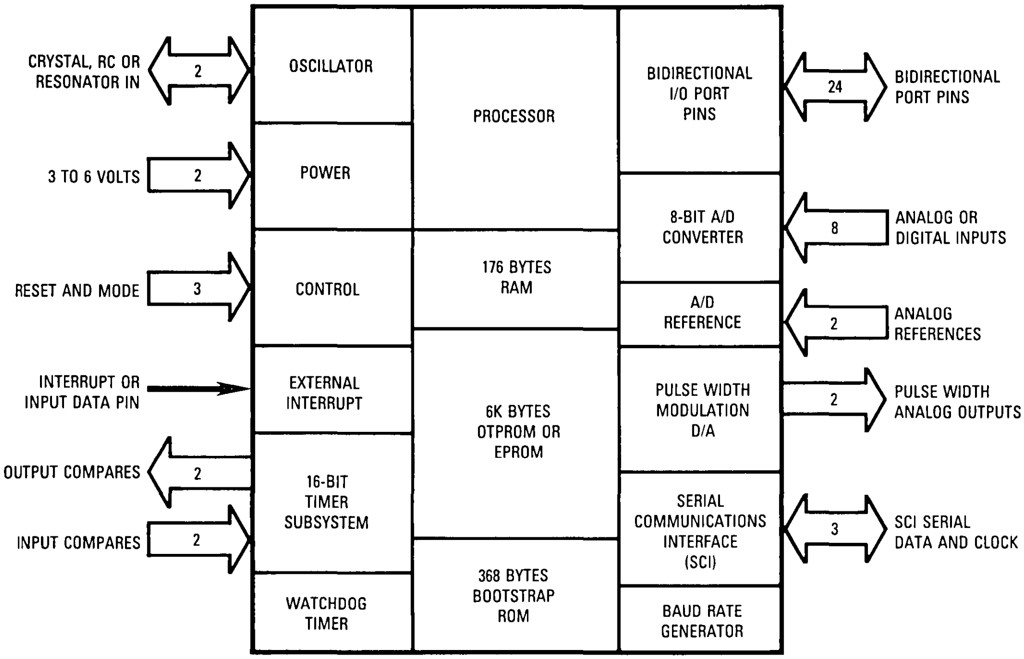
The MC68HC705B5 One Time Programmable ROM (OTPROM) or standard Eraseable Programmable ROM (EPROM) microcontroller unit (MCU) device is similar to the MC68HC05B6 with the following exception. This exception incorporates 6K-bytes of OTPROM or EPROM (refer to block diagram).

The following are some of the features of the MC68HC705B5:

- HCMOS Technology
- 6K Bytes of OTPROM or EPROM
- 176 Bytes of RAM
- 368 Bytes of Bootstrap ROM
- 8-Channel Analog-to-Digital (A/D) Converter
- Serial Communication Interface (SCI)
- 16-Bit Timer Subsystem
- External Timer and SCI Interface
- Fully Static Operation
- Computer Operating Properly (COP) Watchdog Timer
- 24 Bidirectional I/O Lines
- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Option
- 2 MHz Internal Operation at 5 Volts; 1 MHz at 3 Volts
- Single 3- to 6-Volt Supply
- Power-On and External Reset
- 8 × 8 Unsigned Multiply Instruction
- True Bit Manipulation
- Memory Map I/O
- Two Power-Saving Standby Modes (STOP and WAIT)
- 48-Pin DIP or 52-Pin PLCC (OTPROM) Package
- 48-Pin DIP (EPROM Window) Package

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice

BLOCK DIAGRAM


CRYSTAL, RC OR
RESONATOR IN
2
3 TO 6 VOLTS
2
RESET AND MODE
3
INTERRUPT OR
INPUT DATA PIN
OUTPUT COMPARES
2
INPUT COMPARES
2
OSCILLATOR
POWER
CONTROL
EXTERNAL
INTERRUPT
16-BIT
TIMER
SUBSYSTEM
WATCHDOG
TIMER
PROCESSOR
176 BYTES
RAM
6K BYTES
OTPROM OR
EPROM
368 BYTES
BOOTSTRAP
ROM
BIDIRECTIONAL
I/O PORT
PINS
8-BIT A/D
CONVERTER
A/D
REFERENCE
PULSE WIDTH
MODULATION
D/A
SERIAL
COMMUNICATIONS
INTERFACE
(SCI)
BAUD RATE
GENERATOR
24
BIDIRECTIONAL
PORT PINS
8
ANALOG OR
DIGITAL INPUTS
2
ANALOG
REFERENCES
2
PULSE WIDTH
ANALOG OUTPUTS
3
SCI SERIAL
DATA AND CLOCK

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC705C4

Product Preview

One Time Programmable ROM (OTPROM) or Standard Eraseable Programmable Read-Only Memory (EPROM) Microcomputer

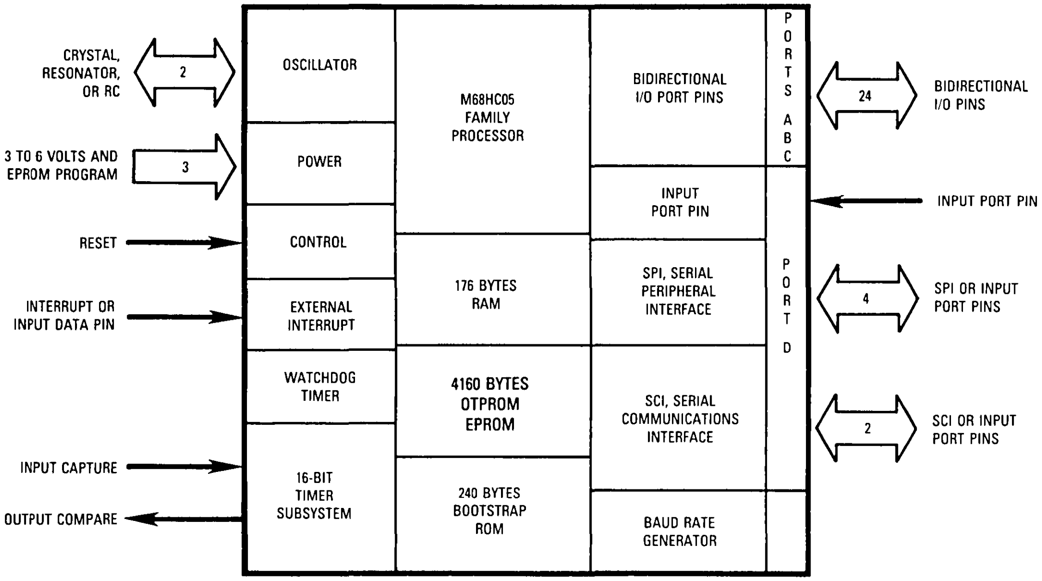
The MC68HC705C4 Microcomputer Unit (MCU) device is similar to the MC68HC05C4 with the following exceptions. These exceptions incorporate 4160 bytes of One Time Programmable Read-Only Memory (OTPROM) or Standard Eraseable Programmable Read-Only Memory (EPROM), Computer Operating Properly (COP) watchdog timer, and 240 bytes of ROM programming bootstrap on-chip (refer to block diagram).

The following are some of the hardware and software features of the MC68HC705C4.

- HCMOS Technology
- Power Saving STOP, WAIT, and Data Retention Modes
- Fully Static Operation
- 4160 Bytes of OTPROM or EPROM
- 176 Bytes of RAM
- 240 Bytes of Bootstrap ROM
- 24 Bidirectional I/O Lines
- 2.1 MHz Internal Operating Frequency at 5 Volts; 1.0 MHz at 3 Volts
- Serial Communications Interface System (SCI)
- Serial Peripheral Interface System (SCI)
- External Timer, SCI, and SPI Interrupts
- Master Reset and Power-On Reset
- Single 3-5 Volt Supply (2-Volt Data Retention Mode)
- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- 8×8 Unsigned Multiply Instruction
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Two Power-Saving Standby Modes, Software Initiated
- Computer Operating Properly (COP) Watchdog Timer
- 16-Bit Timer
- 40-Pin DIP, 44 Lead PLCC (OTPROM) Packages
- 40-Pin DIP (EPROM Window) Package

This document contains information on a new product Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM


CRYSTAL, RESONATOR, OR RC
2
3 TO 6 VOLTS AND EPROM PROGRAM
3
RESET
INTERRUPT OR INPUT DATA PIN
INPUT CAPTURE
OUTPUT COMPARE
OSCILLATOR
POWER
CONTROL
EXTERNAL INTERRUPT
WATCHDOG TIMER
16-BIT TIMER SUBSYSTEM
M68HC05 FAMILY PROCESSOR
176 BYTES RAM
4160 BYTES OTPROM EPROM
240 BYTES BOOTSTRAP ROM
BIDIRECTIONAL I/O PORT PINS
INPUT PORT PIN
SPI, SERIAL PERIPHERAL INTERFACE
SCI, SERIAL COMMUNICATIONS INTERFACE
BAUD RATE GENERATOR
PORTS ABC
PORT D
24
BIDIRECTIONAL I/O PINS
INPUT PORT PIN
4
SPI OR INPUT PORT PINS
2
SCI OR INPUT PORT PINS

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC705C8

Technical Summary

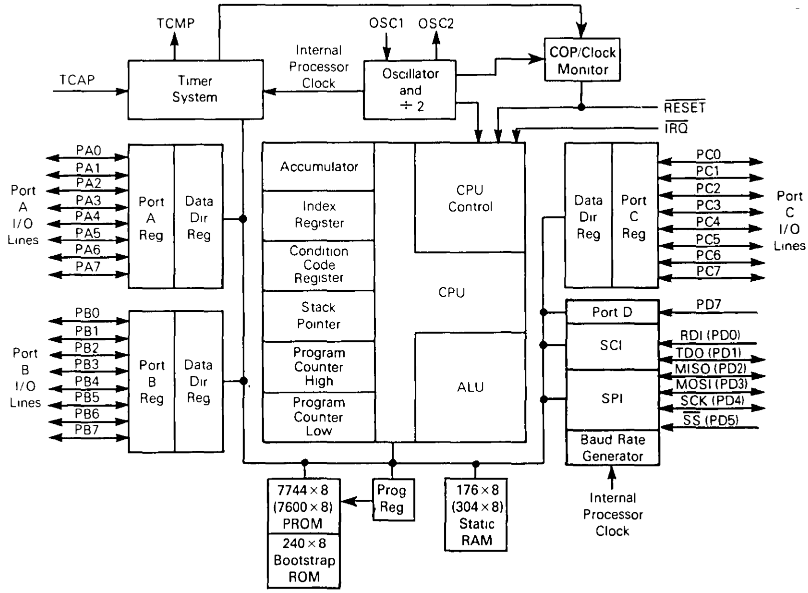
8-Bit Microcontroller Unit

The MC68HC705C8 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers and is available in either one-time programmable ROM (OTPROM) or EPROM versions. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- Selectable Memory Configurations
- Computer Operating Properly (COP) Watchdog Timer
- Clock Monitor
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Bootstrap Capability
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- Software-Programmable External Interrupt Sensitivity

BLOCK DIAGRAM


TCMP
TCAP
Timer System
Internal Processor Clock
OSC1
OSC2
Oscillator and ÷ 2
COP/Clock Monitor
$\overline{\text{RESET}}$
$\overline{\text{IRQ}}$
Port A I/O Lines
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A Reg
Data Dir Reg
Port B I/O Lines
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
$\overline{\text{SS}}$ (PD5)
Baud Rate Generator
Internal Processor Clock
7744 × 8 (7600 × 8) PROM
240 × 8 Bootstrap ROM
Prog Reg
176 × 8 (304 × 8) Static RAM

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{\text{IRQ}}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered

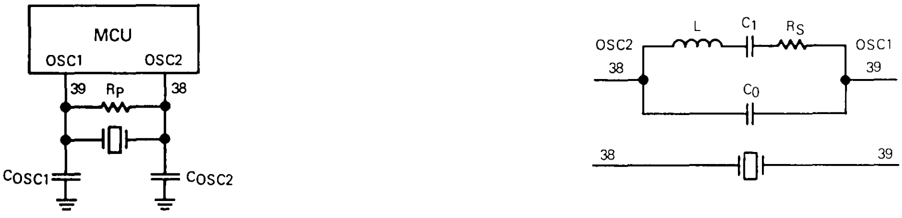
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

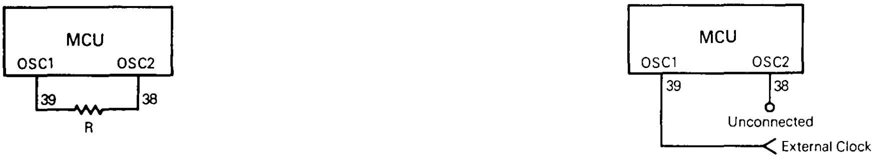
	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections

(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

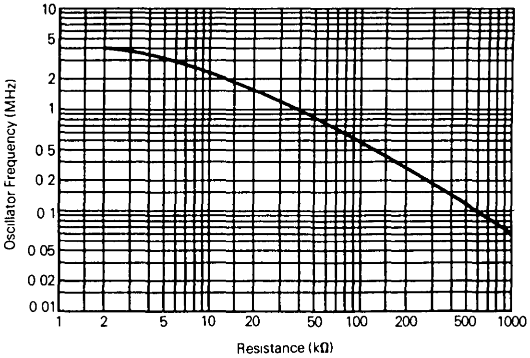


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used as an input to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low. As an output, the $\overline{\text{RESET}}$ pin indicates that an internal MCU failure has been detected.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

V_{PP}

This pin is used to program the OTPROM or EPROM. V_{PP} should be connected to V_{DD} for normal operation.

CAUTION

Do not connect the V_{PP} pin to V_{SS} (ground), or damage to the MCU could result.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

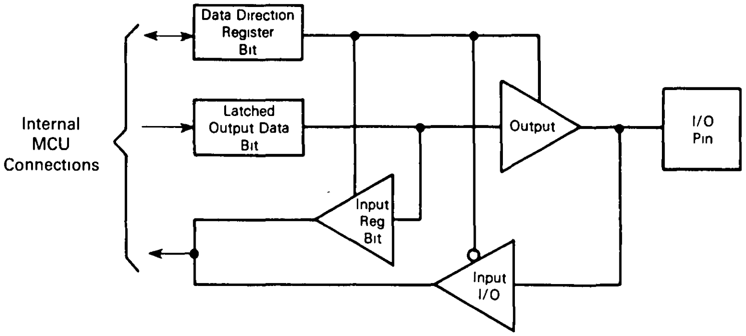


Figure 3. Typical Port I/O Circuit

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers. The locations consist of user PROM, user RAM, bootstrap ROM, control registers, and I/O. User PROM is available as either ultraviolet erasable PROM (EPROM) or one-time programmable read-only memory (OTPROM). The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

The MCU contains four, selectable memory configurations. The first configuration is selected automatically by reset or power-on reset. The memory configuration is selected by the state of the RAM0 and RAM1 bits in the options register ($1FDF). The configurations are selected as follows:

RAM0	RAM1	RAM Bytes	PROM Bytes
0	0	176	7744
1	0	208	7696
0	1	272	7648
1	1	304	7600

Figures 4 through 7 illustrate the four memory configurations.

REGISTERS

The following paragraphs describe the registers that control user options.

Option Register, $1FDF

The option register is used to select the $\overline{IRQ}$ sensitivity, enable the PROM security, and select the memory configuration.

7	6	5	4	3	2	1	0
RAM0	RAM1	0	0	SEC	—	IRQ	0
RESET							
0	0	0	0	U	—	1	0

RAM0 — Random Access Memory Control Bit 0

1 = Maps 32 bytes of RAM into page zero starting at address $0030. Addresses from $0020 to $0030 are reserved. This deletes 48 bytes of PROM that were used at these locations. This bit can be read

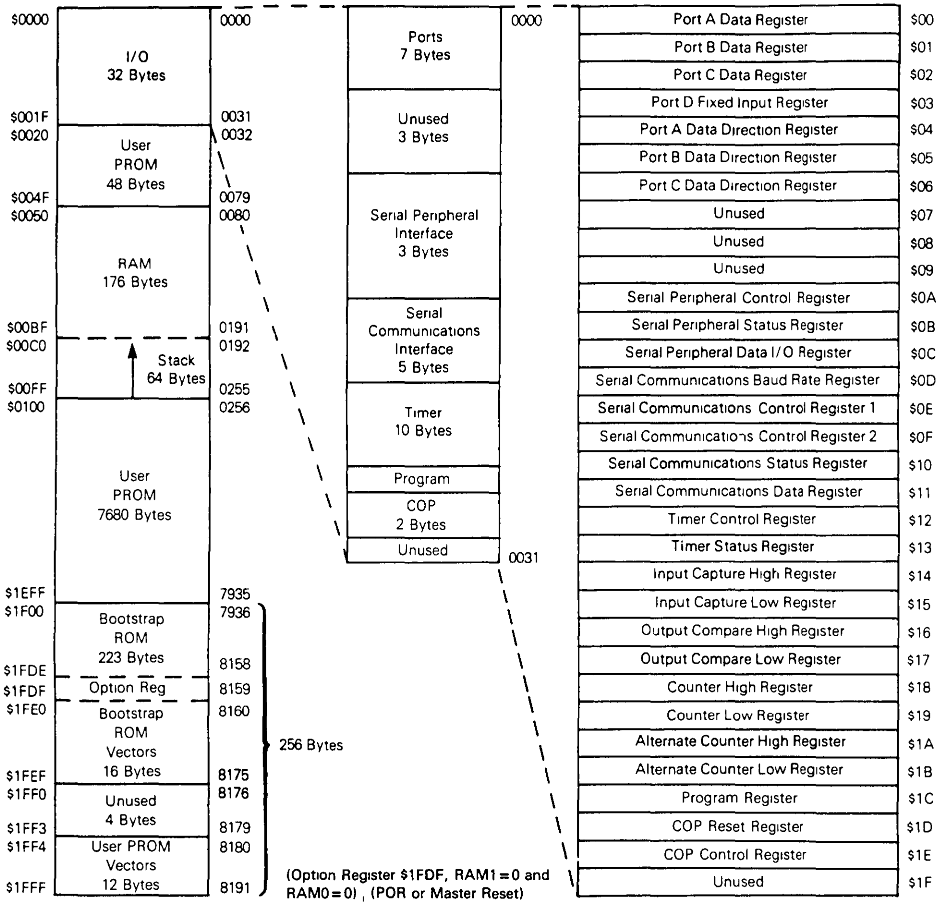

$0000
I/O
32 Bytes
$001F
$0020
User
PROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
User
PROM
7680 Bytes
$1EFF
$1F00
Bootstrap
ROM
223 Bytes
$1FDE
$1FDF
Option Reg
$1FE0
Bootstrap
ROM
Vectors
16 Bytes
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User PROM
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
7935
7936
8158
8159
8160
8175
8176
8179
8180
8191
256 Bytes
(Option Register $1FDF, RAM1=0 and RAM0=0), (POR or Master Reset)
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Program
COP
2 Bytes
Unused
0000
0031
Port A Data Register
$00
Port B Data Register
$01
Port C Data Register
$02
Port D Fixed Input Register
$03
Port A Data Direction Register
$04
Port B Data Direction Register
$05
Port C Data Direction Register
$06
Unused
$07
Unused
$08
Unused
$09
Serial Peripheral Control Register
$0A
Serial Peripheral Status Register
$0B
Serial Peripheral Data I/O Register
$0C
Serial Communications Baud Rate Register
$0D
Serial Communications Control Register 1
$0E
Serial Communications Control Register 2
$0F
Serial Communications Status Register
$10
Serial Communications Data Register
$11
Timer Control Register
$12
Timer Status Register
$13
Input Capture High Register
$14
Input Capture Low Register
$15
Output Compare High Register
$16
Output Compare Low Register
$17
Counter High Register
$18
Counter Low Register
$19
Alternate Counter High Register
$1A
Alternate Counter Low Register
$1B
Program Register
$1C
COP Reset Register
$1D
COP Control Register
$1E
Unused
$1F

Figure 4. Memory Map #1

or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 48 bytes of PROM at location $0030.

RAM1 — Random Access Memory Control Bit 1

1 = Maps 96 bytes of RAM into page zero starting at address $0100. This deletes 96 bytes of PROM that were used at these locations. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 96 bytes of PROM at location $0100.

SEC — Security

1 = Bootloader disabled, MCU operates only in single-chip mode

0 = Security off, bootloader enabled, expanded mode enabled

IRQ — Interrupt Request Bit Sensitivity

1 = $\overline{\mathrm{IRQ}}$ pin is both negative edge- and level-sensitive

0 = $\overline{\mathrm{IRQ}}$ pin is negative edge-sensitive only

IRQ is set only by reset, but can be cleared by software. This bit can only be written once.

Bit 0, 4, 5

Always read zero

Bit 2

Can be either one or zero

Program Register, $1C

The program register ($1C) is used to perform PROM programming.

	7	6	5	4	3	2	1	0
	0	0	0	0	0	LAT	0	PGM
RESET:	0	0	0	0	0	0	0	0

LAT — Latch Enable

1 = Enables PROM data and address bus latches for programming or erasing on the next byte write cycle.

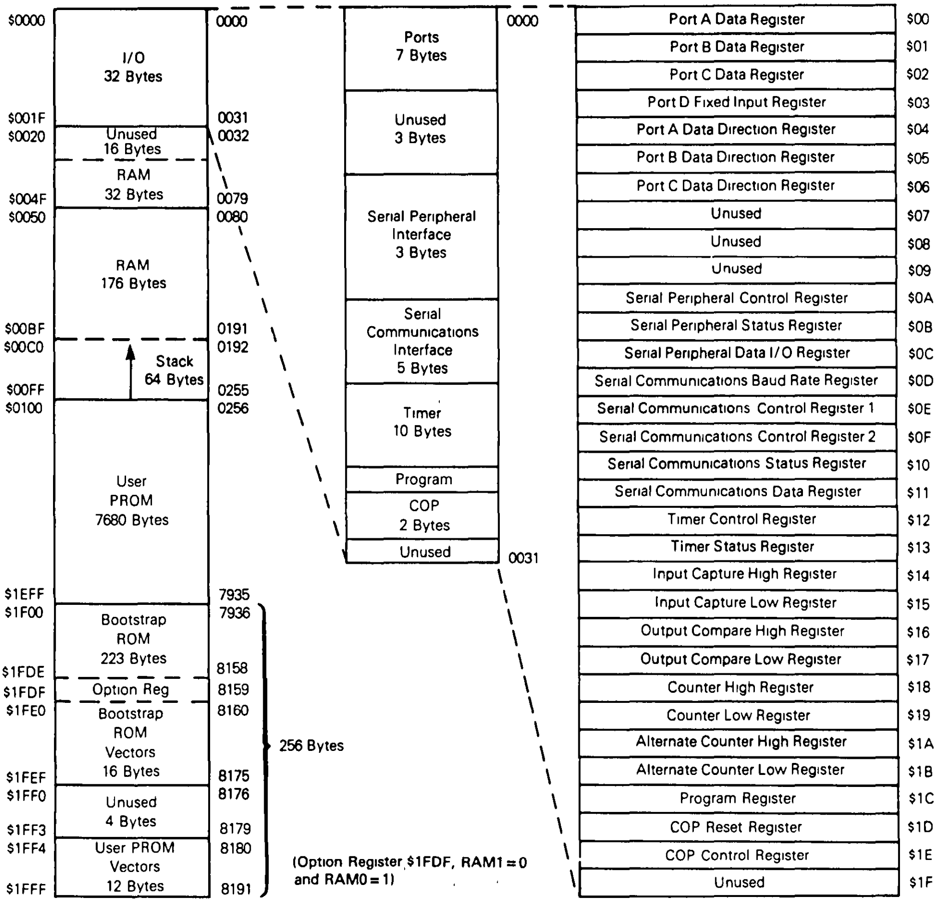

$0000
$001F
$0020
$004F
$0050
$00BF
$00C0
$00FF
$0100
$1EFF
$1F00
$1FDE
$1FDF
$1FE0
$1FEF
$1FF0
$1FF3
$1FF4
$1FFF
I/O
32 Bytes
Unused
16 Bytes
RAM
32 Bytes
RAM
176 Bytes
Stack
64 Bytes
User
PROM
7680 Bytes
Bootstrap
ROM
223 Bytes
Option Reg
Bootstrap
ROM
Vectors
16 Bytes
Unused
4 Bytes
User PROM
Vectors
12 Bytes
0000
0031
0032
0079
0080
0191
0192
0255
0256
7935
7936
8158
8159
8160
8175
8176
8179
8180
8191
256 Bytes
(Option Register $1FDF, RAM1 = 0
and RAM0 = 1)
0000
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Program
COP
2 Bytes
Unused
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Program Register $1C
COP Reset Register $1D
COP Control Register $1E
Unused $1F

Figure 5. Memory Map #2

0 = Latch disabled. PROM data and address buses are unlatched for normal CPU operations.

This bit is both readable and writable.

PGM — Program

1 = Applies Vpp power to the PROM for programming.

0 = Vpp power off

If LAT is cleared, PGM cannot be set.

Bits 1, 3–7 — Not used

Always read zero

Erasing

OTPROM MCU devices are shipped in an erased state and cannot be erased. Electrical erasing procedures cannot be performed on these devices.

EPROM devices can be erased by exposure to a high-intensity ultraviolet (UV) light with a wavelength of 2537 Angstrom. The recommended dose (UV intensity times exposure time) is 15Ws/cm^2. UV lamps should be used without shortwave filters, and the EPROM device should be positioned about one inch from the UV lamps.

OTPROM/EPROM Programming

Figure 8 illustrates the programming sequence.

The OTPROM or EPROM programming technique can be used to load a user program into the MCU. A user program contained in external EPROM can be copied into the internal PROM of the MC68HC705C8.

NOTE

The SEC bit in the option register disables the bootstrap loader.

The MCU device is inserted into the circuit shown in Figure 9. A programming routine is selected via switches S1 through S4, and V_{DD} and V_{PP} applied to the circuit. Switch S5 changes the MCU from RESET to RUN mode,

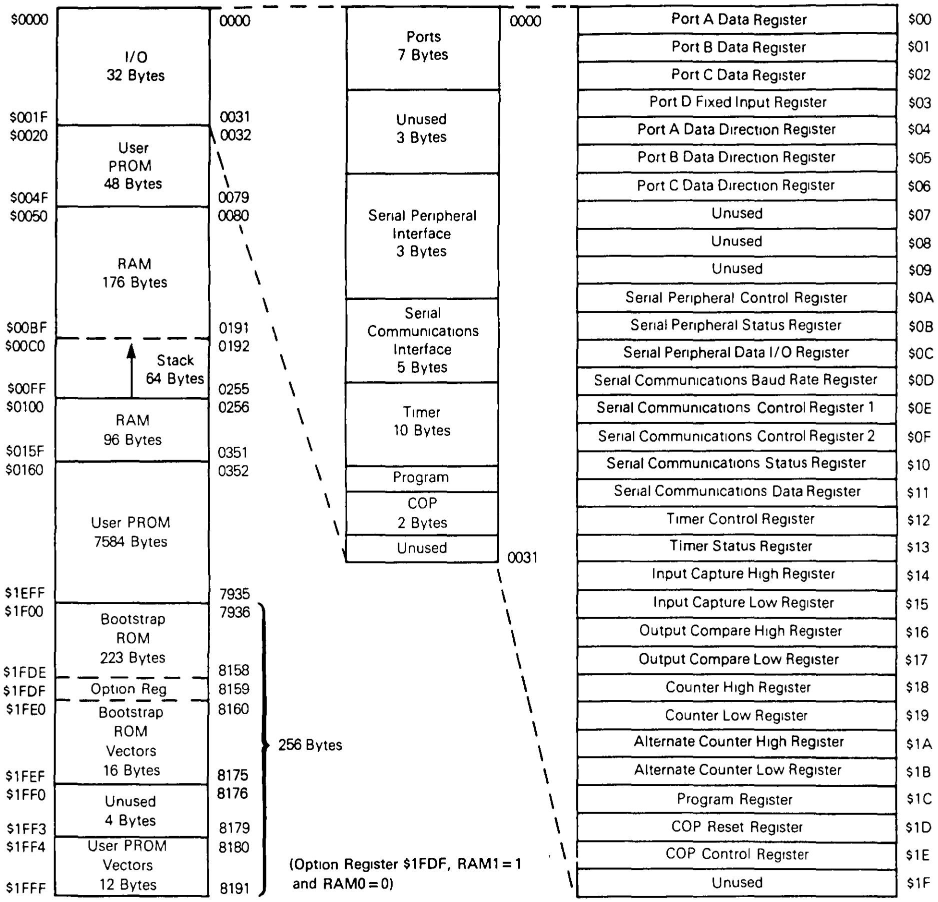

$0000
I/O
32 Bytes
$001F
$0020
User PROM 48 Bytes
$004F
$0050
RAM 176 Bytes
$00BF
$00C0
Stack 64 Bytes
$00FF
$0100
RAM 96 Bytes
$015F
$0160
User PROM 7584 Bytes
$1EFF
$1F00
Bootstrap ROM 223 Bytes
$1FDE
$1FDF
Option Reg
$1FE0
Bootstrap ROM Vectors 16 Bytes
$1FEF
$1FF0
Unused 4 Bytes
$1FF3
$1FF4
User PROM Vectors 12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
0351
0352
7935
7936
8158
8159
8160
8175
8176
8179
8180
8191
256 Bytes
(Option Register $1FDF, RAM1 = 1 and RAM0 = 0)
Ports 7 Bytes
Unused 3 Bytes
Serial Peripheral Interface 3 Bytes
Serial Communications Interface 5 Bytes
Timer 10 Bytes
Program
COP 2 Bytes
Unused
0000
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Program Register $1C
COP Reset Register $1D
COP Control Register $1E
Unused $1F

Figure 6. Memory Map #3

control transfers to the bootstrap ROM, and the selected routine is executed.

The EPROM programming sequence of events is as follows:

1. Place S5 in the RESET position.
2. Select routine with S1 through S5.
3. Apply V_{DD} and V_{PP} to the circuit.
4. Place S5 in the RUN position.
5. Programming routine is executed.
6. Place S5 in the RESET position.
7. Remove V_{DD} and V_{PP}, or select and run new routine.

Once in bootstrap mode, the mode switch settings establish the routine to be executed. The routines are as follows:

Program and verify PROM
Verify PROM contents
Secure PROM
Load program in RAM and execute
Dump PROM contents
Execute program in RAM

Program and Verify PROM

The program and verify routine copies the contents of an external 8K EPROM into the MCU PROM, with direct correspondence between the addresses. Memory addresses in the MCU that are not implemented in PROM are skipped. Unprogrammed EPROM addresses should contain $00 bytes to speed up the programming process. During programming, the PROGRAMMING LED (DS2) lights. After programming, DS2 turns off and verification begins. If the contents of the external EPROM and MCU internal PROM exactly match, the VERIFIED LED (DS1) lights. If a discrepancy is detected, the routine stops and the error address location is placed on the external memory address bus.

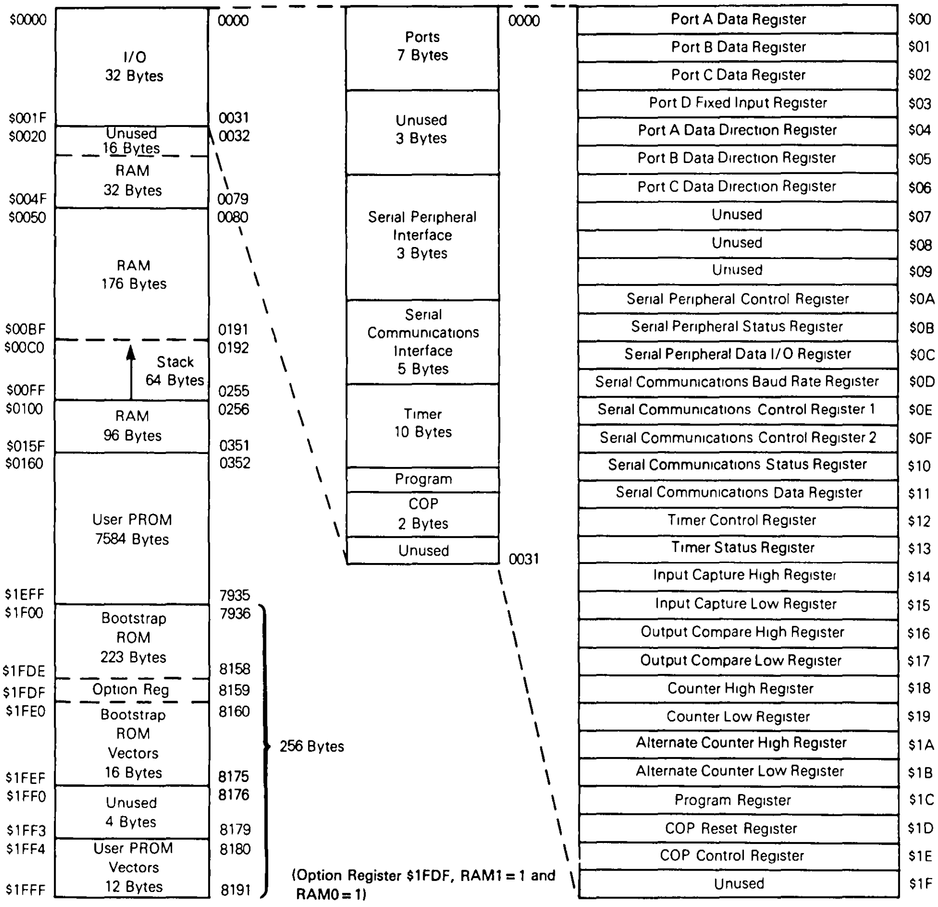

$0000
I/O
32 Bytes
$001F
$0020
Unused
16 Bytes
RAM
32 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64 Bytes
$00FF
$0100
RAM
96 Bytes
$015F
$0160
User PROM
7584 Bytes
$1EFF
$1F00
Bootstrap
ROM
223 Bytes
$1FDE
$1FDF
Option Reg
$1FE0
Bootstrap
ROM
Vectors
16 Bytes
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User PROM
Vectors
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
0351
0352
7935
7936
8158
8159
8160
8175
8176
8179
8180
8191
256 Bytes
(Option Register $1FDF, RAM1 = 1 and RAM0 = 1)
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
Program
COP
2 Bytes
Unused
0000
0031
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Port D Fixed Input Register $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
Unused $07
Unused $08
Unused $09
Serial Peripheral Control Register $0A
Serial Peripheral Status Register $0B
Serial Peripheral Data I/O Register $0C
Serial Communications Baud Rate Register $0D
Serial Communications Control Register 1 $0E
Serial Communications Control Register 2 $0F
Serial Communications Status Register $10
Serial Communications Data Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
Program Register $1C
COP Reset Register $1D
COP Control Register $1E
Unused $1F

Figure 7. Memory Map #4

Verify PROM Contents

The verify PROM routine is normally entered automatically after the PROM is programmed. Direct entry of this routine causes the PROM contents to be compared to the contents of external memory locations at the same addresses. Both DS1 and DS2 are turned off until verification is complete. After verification, DS1 lights. If verification fails, the routine halts with the failing address in the external memory address bus.

Secure PROM

This routine is used after the PROM is successfully programmed and verified. Only the SEC bit in the option register ($1FDF) is programmed, and the VERIFIED LED (DS1) lights to indicate the end of the routine. This does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the proper LED does not light, the PROM has been properly secured.

Load Program in RAM and Execute

In the load program in RAM and execute routine, user programs are loaded via the SCI port, and then executed. Data is loaded sequentially, starting at address $0050. After the last byte is loaded, control is transferred to the RAM program starting at location $0051. The first byte loaded is the count of the total number of bytes in the program, plus the count byte. The program starts at the second location in RAM. During initialization, the SCI is configured for NRZ data format (idle line, start bit, 8 data bytes, and stop bit). The baud rate is 4800 with a 2-MHz crystal.

Execution can be held off by setting the byte count to a value greater than the number of bytes to be loaded. After loading the last byte, the firmware waits for more data. At this point, S5 can be placed in the RESET position, which resets the MCU with the RAM data intact. All other routines can be entered, including the one to execute program in RAM, by selecting the routine desired and switching S5 to RUN.

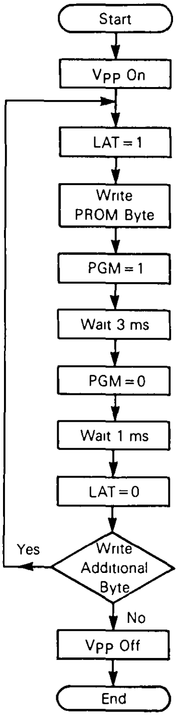


Figure 8. OTPROM/EPROM Programming

Dump PROM Contents

In the dump PROM contents routine, the PROM contents are dumped sequentially to the SCI output. The first location sent is $0020, and the last location sent is $1FFF. Unused locations are skipped so that no gaps exist in the data stream. The external memory address lines indicate the current location being sent. Data is sent in NRZ format, as in the load program in RAM routine.

Execute Program in RAM

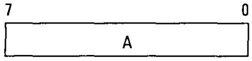
This routine allows the MCU to transfer control to a program previously loaded in RAM. This program is executed once bootstrap mode is entered, if switch S4 is activated, without any firmware initialization. The program must start at location $0051 to be compatible with the load program in RAM routine.

REGISTERS

The MCU contains the registers described in the following paragraphs.

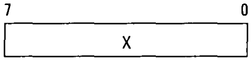
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



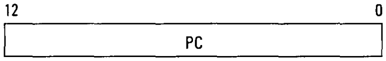
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

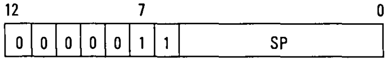
The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

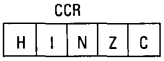
The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

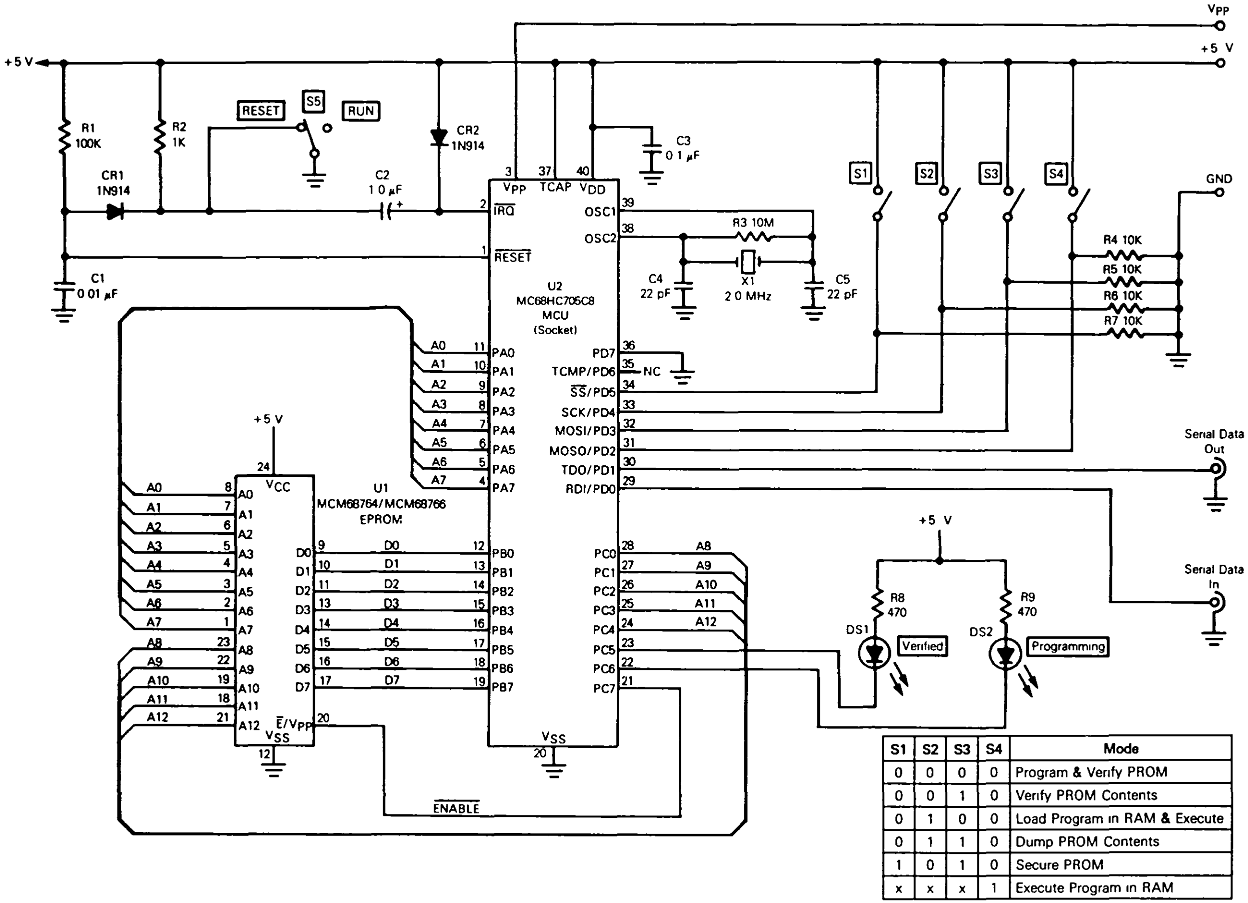


Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is


VPP
+5 V
+5 V
GND
R1 100K
R2 1K
CR1 1N914
RESET
S5
RUN
CR2 1N914
C2 1.0 µF
C1 0.01 µF
C3 0.1 µF
U2 MC68HC705C8 MCU (Socket)
VPP
TCAP
VDD
IRQ
RESET
OSC1
OSC2
R3 10M
X1 2.0 MHz
C4 22 pF
C5 22 pF
S1
S2
S3
S4
R4 10K
R5 10K
R6 10K
R7 10K
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PD7
TCMP/PD6
NC
SS/PD5
SCK/PD4
MOSI/PD3
MOSO/PD2
TDO/PD1
RDI/PD0
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
VSS
Serial Data Out
Serial Data In
U1 MCM68764/MCM68766 EPROM
VCC
E/VPP
VSS
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
D0
D1
D2
D3
D4
D5
D6
D7
ENABLE
R8 470
R9 470
DS1
DS2
Verified
Programming

S1	S2	S3	S4	Mode
0	0	0	0	Program & Verify PROM
0	0	1	0	Verify PROM Contents
0	1	0	0	Load Program in RAM & Execute
0	1	1	0	Dump PROM Contents
1	0	1	0	Secure PROM
x	x	x	1	Execute Program in RAM

Figure 9. PROM Programming Circuit

set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

RESETS

The MCU can be reset in the following four ways:

1. An internal, power-on condition.
2. An external, active-low input on the $\overline{\text{RESET}}$ pin.
3. An internal computer operating properly (COP) watchdog timer reset condition.
4. An internal clock monitor reset condition.

POWER-ON RESET (POR)

An internal reset is generated on powerup to allow the internal clock generator to stabilize. The power-on reset is strictly for power turnon conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 4064 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

Computer Operating Properly (COP) Watchdog Timer Reset

The MCU includes a COP watchdog timer to help protect against software failures. Once the COP is enabled, a COP reset sequence must be executed on a periodic basis so the COP does not time out. Since the COP timer uses the internal bus clock, a clock monitor is included to guard against clock failure.

The COP reset register ($1D) and the COP control register ($1E) shown below are used to control the COP watchdog timer and clock monitor functions.

COP Reset Register (1D)

7	6	5	4	3	2	1	0
RESET:							
0	0	0	0	0	0	0	0

The sequence required to reset the COP timer is as follows:

Write $55 to the COP reset register
Write $AA to the COP reset register

Both write operations must occur in the order listed, but any number of instructions may be executed between the two write operations. The elapsed time between software resets must not be greater than the COP timeout period. Reading the COP reset register does not return valid data and does not affect the watchdog timer.

COP Control Register (1E)

7	6	5	4	3	2	1	0
0	0	0	COPF	CME	COPE	CM1	CM0
RESET.							
0	0	0	0	0	0	0	0

COPF — Computer Operating Properly
1 = COP or clock monitor reset has occurred
0 = No COP or clock monitor reset has occurred
Reading the COP control register clears COPF

CME — Clock Monitor Enable
1 = Clock monitor enabled
0 = Clock monitor disabled
CME is readable and writable at any time

COPE — Computer Operating Properly Enable
1 = COP timeout enabled
0 = COP timeout disabled

CM1 — Computer Operating Properly Mode 1
Used in conjunction with CM0 to establish the COP timeout period. CM1 can be read and set anytime, but is cleared only by reset. See Table 2.

CM0 — Computer Operating Properly Mode 0
Used in conjunction with CM1 to establish the COP timeout period. CM0 can be read and set anytime, but is cleared only by reset. See Table 2.

Bits 5–7 — Not used
Always read zero

Table 2. COP Timeout Period

CM1	CM0	$E/2^{15}$ Divided By	XTAL = 4.0 MHz, E = 2.0 MHz	XTAL = 3.5795 MHz, E = 1.7897 MHz	XTAL = 2.0 MHz, E = 1.0 MHz	XTAL = 1.0 MHz, E = 0.5 MHz
0	0	1	16.38 ms	18.31 ms	32.77 ms	65.54 ms
0	1	4	65.54 ms	73.24 ms	131.07 ms	262.14 ms
1	0	16	262.14 ms	292.95 ms	524.29 ms	1.048 s
1	1	64	1.048 s	1.172 s	2.097 s	4.194 s

Clock Monitor Reset

When the CME bit in the COP control register is set, the clock monitor detects the absence of the internal bus clock for a certain period of time. The timeout period depends on processing parameters and varies from 5 to 100 µs, which implies that systems using a bus clock rate of 200 kHz or less should not use the clock monitor function.

If a slow or absent clock is detected, the clock monitor causes a system reset. The reset is issued to the external system for four bus cycles via the bidirectional $\overline{\text{RESET}}$ pin.

Special consideration is required when using the STOP instruction with the clock monitor. Since STOP causes the system clocks to halt, the clock monitor issues a system reset when STOP is executed.

The clock monitor is a useful backup to the COP watchdog timer. Since the watchdog timer requires a clock to function, the timer will not indicate any failure if the system clocks fail. The clock monitor would detect such a failure and force the MCU to a reset state. Clocks are not required for the MCU to reach a reset condition, although clocks are required to sequence through reset back to the run condition.

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts ($\overline{\text{IRQ}}$, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 10.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 11 for the reset and interrupt instruction processing sequence.

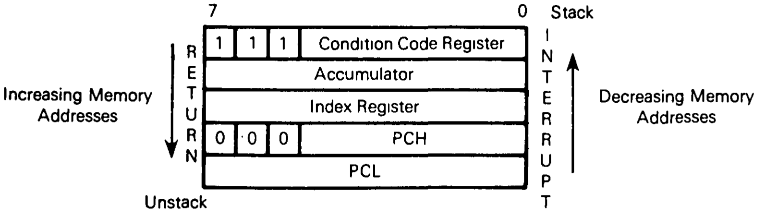
TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a software option. See **Option Register** for more information. Figure 12 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 10. Interrupt Stacking Order

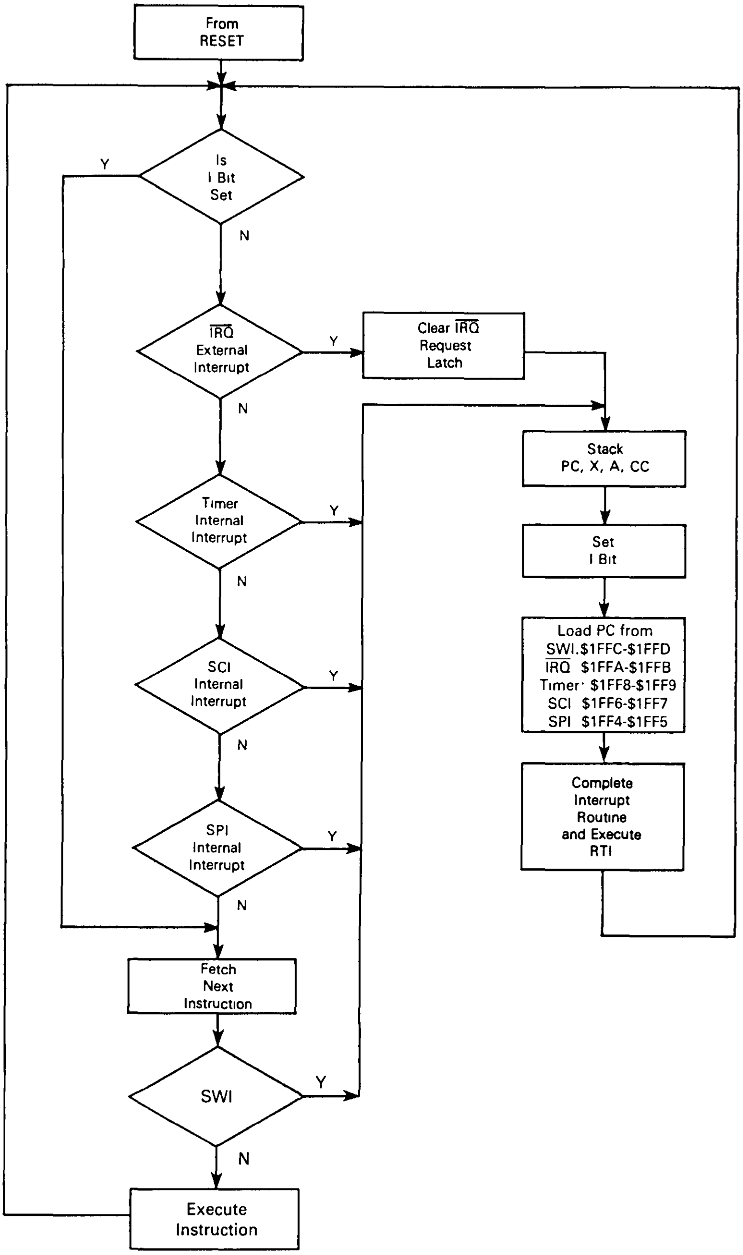
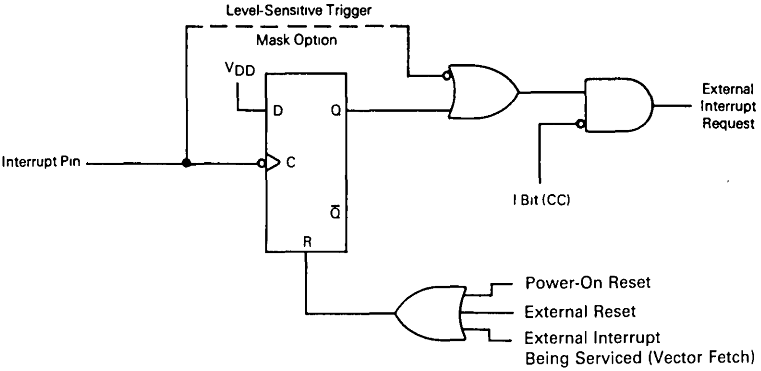
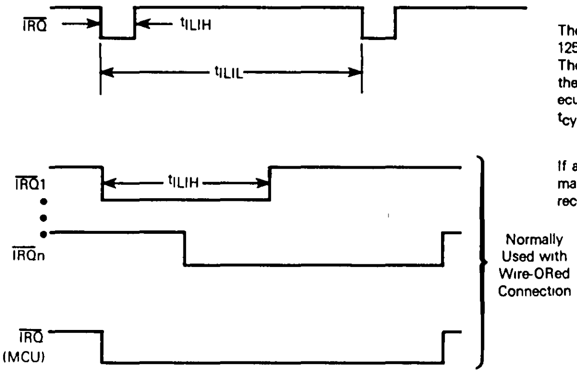

From RESET
Is I Bit Set
Y
N
IRQ External Interrupt
Y
Clear IRQ Request Latch
N
Stack PC, X, A, CC
Timer Internal Interrupt
Y
Set I Bit
N
Load PC from
SWI.$1FFC-$1FFD
IRQ $1FFA-$1FFB
Timer· $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI $1FF4-$1FF5
SCI Internal Interrupt
Y
N
Complete Interrupt Routine and Execute RTI
SPI Internal Interrupt
Y
N
Fetch Next Instruction
SWI
Y
N
Execute Instruction

Figure 11. Reset and Interrupt Processing Flowchart


Level-Sensitive Trigger
Mask Option
V_{DD}
D
Q
C
$\overline{Q}$
R
Interrupt Pin
External Interrupt Request
I Bit (CC)
Power-On Reset
External Reset
External Interrupt Being Serviced (Vector Fetch)

(a) Interrupt Internal Function Diagram


$\overline{IRQ}$
t_{ILIH}
t_{ILIL}
$\overline{IRQ}1$
$\overline{IRQ}n$
$\overline{IRQ}$ (MCU)
Normally Used with Wire-ORed Connection

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 12. External Interrupt

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 13).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI during STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{IRQ}$ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

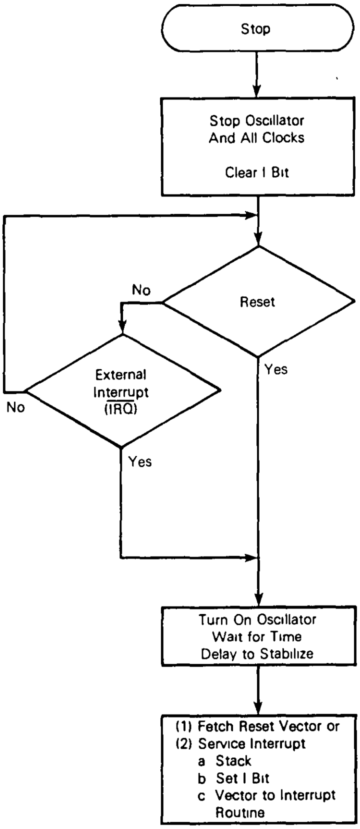

Stop
Stop Oscillator And All Clocks
Clear I Bit
Reset
No
Yes
External Interrupt ($\overline{IRQ}$)
No
Yes
Turn On Oscillator Wait for Time Delay to Stabilize
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt Routine

Figure 13. STOP Function Flowchart

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the $\overline{IRQ}$ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the $\overline{IRQ}$ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active (refer to Figure 14). An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{RESET}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 15 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

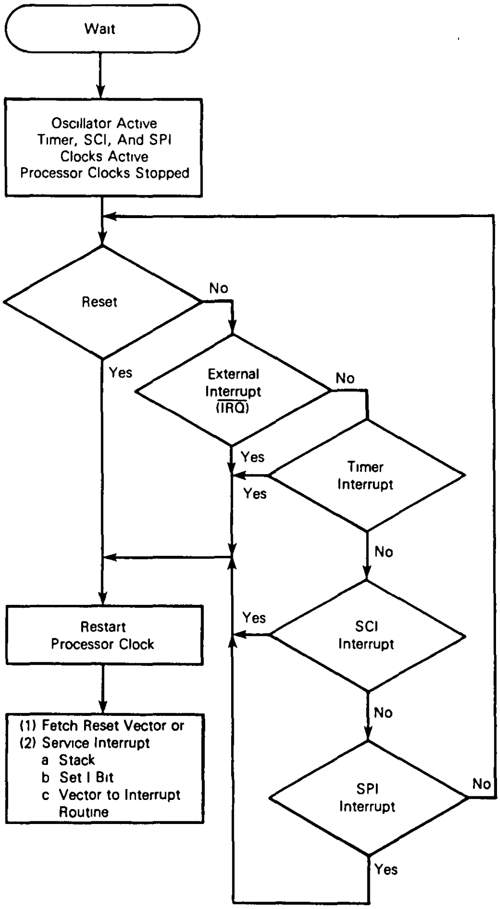

Wait
Oscillator Active
Timer, SCI, And SPI
Clocks Active
Processor Clocks Stopped
Reset
No
Yes
External
Interrupt
($\overline{IRQ}$)
No
Yes
Timer
Interrupt
Yes
No
Restart
Processor Clock
SCI
Interrupt
Yes
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt
Routine
SPI
Interrupt
No
Yes

Figure 14. WAIT Function Flowchart

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18–$19 (counter register) or $1A–$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without

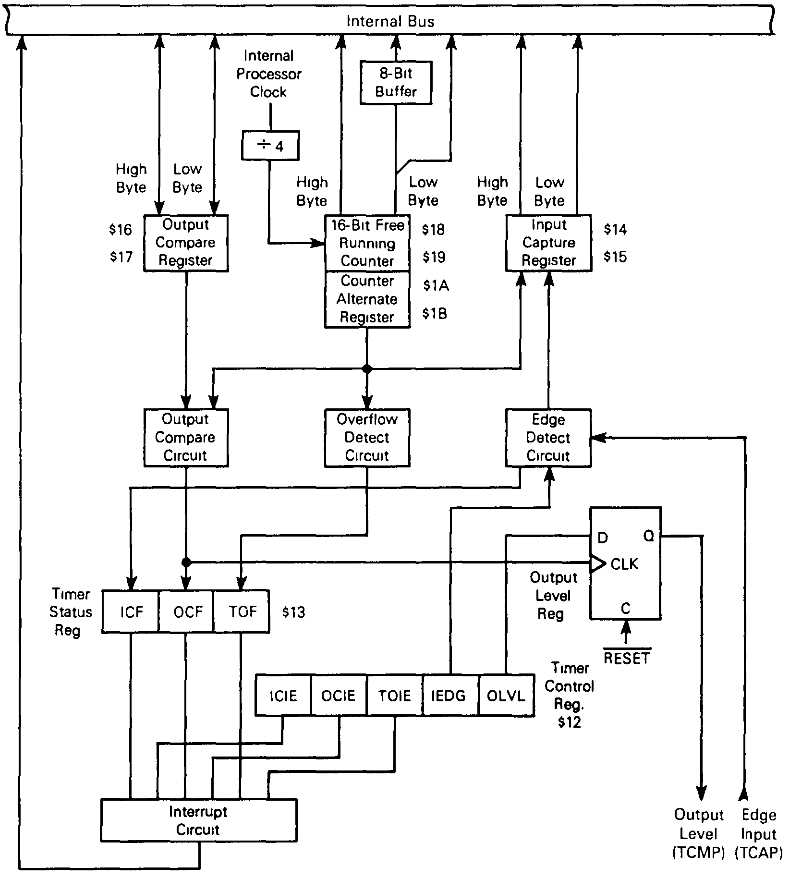

Internal Bus
Internal Processor Clock
8-Bit Buffer
÷ 4
High Byte
Low Byte
High Byte
Low Byte
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
$18
$19
Counter Alternate Register
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
Timer Control Reg. $12
Interrupt Circuit
Output Level (TCMP)
Edge Input (TCAP)

Figure 15. Timer Block Diagram

the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
OCIE — Output Compare Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
TOIE — Timer Overflow Interrupt Enable
  1 = Interrupt enabled
  0 = Interrupt disabled
IEDG — Input Edge
  Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
  1 = Positive edge
  0 = Negative edge
  Reset does not affect te IEDG bit (U = unaffected).
OLVL — Output Level
  Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
  1 = High output
  0 = Low output
Bits 2, 3, and 4 — Not used
  Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
  1 = Flag set when selected polarity edge is sensed by input capture edge detector
  0 = Flag cleared when TSR and input capture low register ($15) are accessed
OCF — Output Compare Flag
  1 = Flag set when output compare register contents match the free-running counter contents
  0 = Flag cleared when TSR and output compare low register ($17) are accessed
TOF — Timer Overflow Flag
  1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
  0 = Flag cleared when TSR and counter low register ($19) are accessed
Bits 0–4 — Not used
  Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 16.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 17); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise

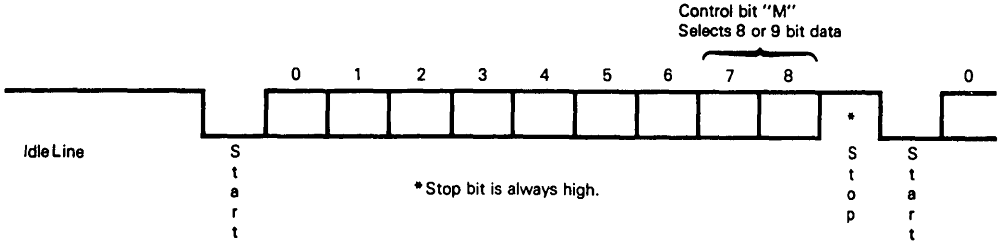


Figure 16. Data Format

flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 17. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates, which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET	U	U	U	U	U	U	U	U

As shown in Figure 17, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $OE

The SCCR1 provides control bits that determine word length and select the wake-up method.

	7	6	5	4	3	2	1	0
	R8	T8	—	M	WAKE	—	—	—
RESET:	U	U	—	U	U	—	—	—

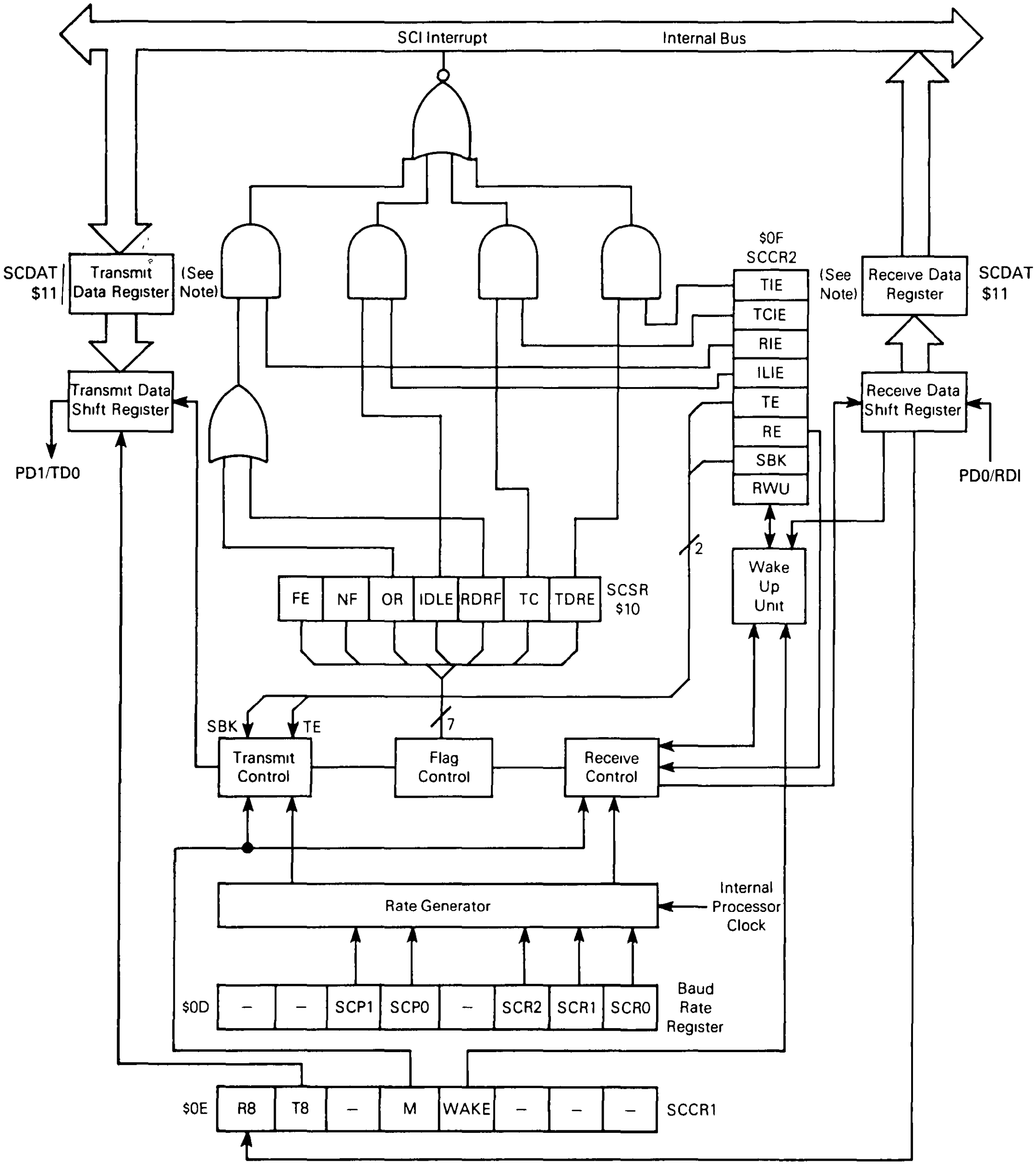
R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most significant bit)
0 = Idle line condition

Bits 0–2, and 5 — Not used
Can read either one or zero


SCI Interrupt
Internal Bus
SCDAT $11
Transmit Data Register
(See Note)
Transmit Data Shift Register
PD1/TD0
$0F SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(See Note)
Receive Data Register
SCDAT $11
Receive Data Shift Register
PD0/RDI
Wake Up Unit
2
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $10
SBK
TE
7
Transmit Control
Flag Control
Receive Control
Rate Generator
Internal Processor Clock
$0D
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
Baud Rate Register
$0E
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 17. SCI Block Diagram

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $0F

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:							
0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable

1 = SCI interrupt enabled

0 = TDRE interrupt disabled

TCIE — Transmit Complete Interrupt Enable

1 = SCI interrupt enabled

0 = TC interrupt disabled

RIE — Receive Interrupt Enable

1 = SCI interrupt enabled

0 = RDRF and OR interrupts disabled

ILIE — Idle Line Interrut Enable

1 = SCI interrupt enabled

0 = Idle interrupt disabled

TE — Transmit Enable

1 = Transmit shift register output is applied to the TDO line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.

0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TDO line becomes a high-impedance line.

RE — Receive Enable

1 = Receiver shift register input is applied to the RDI line.

0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.

RWU — Receiver Wake-Up

1 = Places receiver in sleep mode and enables wake-up function

0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)

SBK — Send Break

1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.

0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET:							
1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty

1 = TDR contents transferred to the transmit data shift register

0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1), followed by a write to the TDR.

TC — Transmit Complete

1 = Indicates end of data frame, preamble, or break condition has occurred

0 = TC bit cleared by reading the SCSR (with TC = 1), followed by a write to the TDR

RDRF — Receive Data Register (RDR) Full

1 = Receive data shift register contents transferred to the RDR

0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR

IDLE — Idle Line Detect

1 = Indicates receiver has detected an idle line

0 = IDLE is cleared by reading the SCSR (with IDLE = 1), followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error

1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.

0 = OR is cleared by reading the SCSR (with OR = 1), followed by a read of the RDR.

NF — Noise Flag

1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.

0 = NF is cleared by reading the SCSR (with NF = 1), followed by a read of the RDR.

FE — Framing Error

1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If

received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.

0 = NF is cleared by reading the SCSR (with FE = 1), followed by a read of the RDR.

Bit 0 — Not used

Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

7	6	5	4	3	2	1	0
—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET:							
—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0

SCP1 — SCI Prescaler Bit 1

Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 2.

SCR0 — SCI Baud Rate Bit 0

SCR1 — SCI Baud Rate Bit 1

SCR2 — SCI Baud Rate Bit 2

Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 3.

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided-by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600-Hz baud rate is required with a 2.4576-MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	0		4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 18) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{\text{SS}}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{\text{SS}}=1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 19, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{\text{SS}}$) input line selects a slave device. The $\overline{\text{SS}}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{\text{SS}}$ line on the master must be tied high; if the $\overline{\text{SS}}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{\text{SS}}$ with SCK. In this clock phase mode, $\overline{\text{SS}}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{\text{SS}}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{\text{SS}}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{\text{SS}}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 20. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based

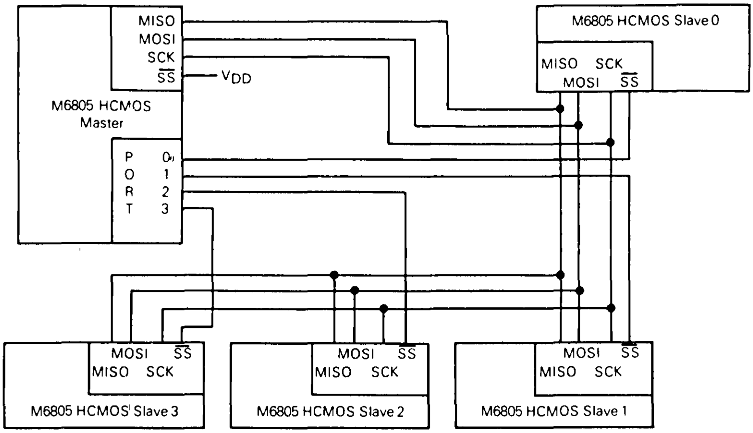


Figure 18. Master-Slave System Configuration

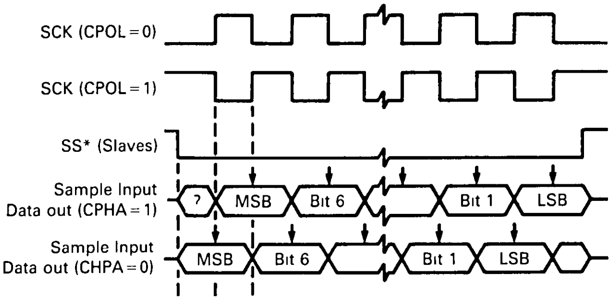

SCK (CPOL=0)
SCK (CPOL=1)
SS* (Slaves)
Sample Input
Data out (CPHA=1)
?
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA=0)
MSB
Bit 6
Bit 1
LSB

Figure 19. Data Clock Timing Diagram

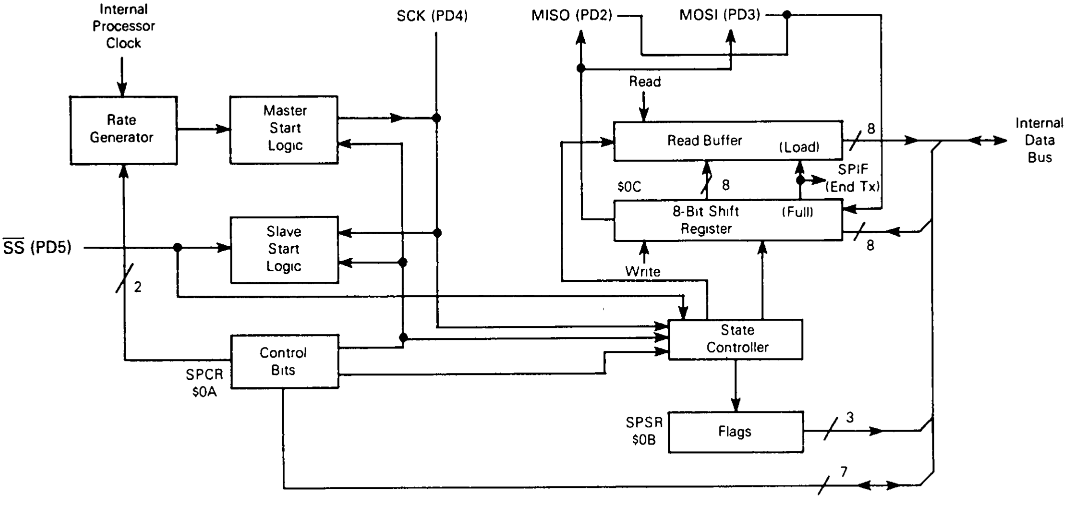

Internal Processor Clock
Rate Generator
Master Start Logic
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Read
Read Buffer
(Load)
8
Internal Data Bus
SPIF
(End Tx)
$0C
8
8-Bit Shift Register
(Full)
8
Write
SS (PD5)
Slave Start Logic
2
State Controller
Control Bits
SPCR
$0A
SPSR
$0B
Flags
3
7

Figure 20. SPI Block Diagram

on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the SS pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 21 illustrates the MOSI, MISO, SCK, and SS master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

7	6	5	4	3	2	1	0
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:							
0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled

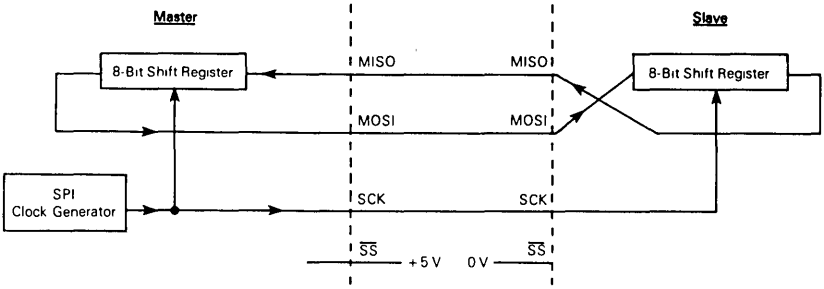


Figure 21. SPI Master-Slave Interconnections

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity

Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase

Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits

Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 - Not used

Can read either one or zero

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—
RESET.							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device. (If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.

WCOL — Write Collision
1 = Indicates an attempt is made to write to SPDR while data transfer is in process.
0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict.
0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 0–3, and 5 — Not used

Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read/write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

<table>
<tr><td>Operation</td><td colspan="4">X:A X*A</td></tr>
<tr><td>Description</td><td colspan="4">Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register</td></tr>
<tr><td>Condition Codes</td><td colspan="4">H: Cleared
I: Not affected
N: Not affected
Z: Not affected
C: Cleared</td></tr>
<tr><td>Source</td><td colspan="4">MUL</td></tr>
<tr><td rowspan="2">Form(s)</td><td>Addressing Mode</td><td>Cycles</td><td>Bytes</td><td>Opcode</td></tr>
<tr><td>Inherent</td><td>11</td><td>1</td><td>$42</td></tr>
</table>

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510

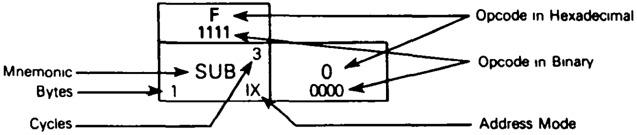
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Programming Voltage	V_{PP}	V_{DD} −0.3 to 16.0	V
Bootstrap Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to $2 \times V_{DD} + 0.3$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC705C8P, FN, S MC68HC705C8CP, CFN, CS	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Plastic Leaded Chip Carrier (PLCC) Cerdip	θ_{JA}	 60 70 60	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

$V_{DD} = 4.5$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0, PD5, PD7	1.9 kΩ	2 26 kΩ	200 pF

$V_{DD} = 3.0$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ 6.32 kΩ	50 pF	
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF

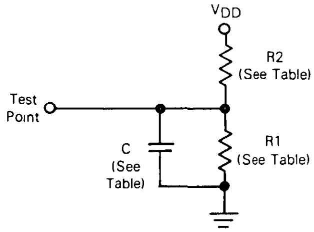


Figure 22. Equivalent Test Load

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leqslant 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 23) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 24)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 25) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run Wait Stop 25°C 0° to 70°C (Standard) −40° to +85°C	I_{DD}	 — — — — —	 5 1.6 2.0 — —	 TBD TBD TBD TBD TBD	 mA mA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 TBD TBD	pF

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 23) (I_{Load} = 0.4 mA) PD1-PD4 (see Figure 24)	V_{OH}	 $V_{DD} - 0.3$ $V_{DD} - 0.3$	 — —	 — —	V
Output Low Voltage (see Figure 25) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run Wait Stop 25°C 0° to 70°C (Standard) −40° to +85°C	I_{DD}	 — — — — —	 2.0 0.5 1.0 — —	 TBD TBD TBD TBD TBD	 mA mA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}. Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}. All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. Extended temperature (−40° to +85°C,) and a 25°C only version are available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

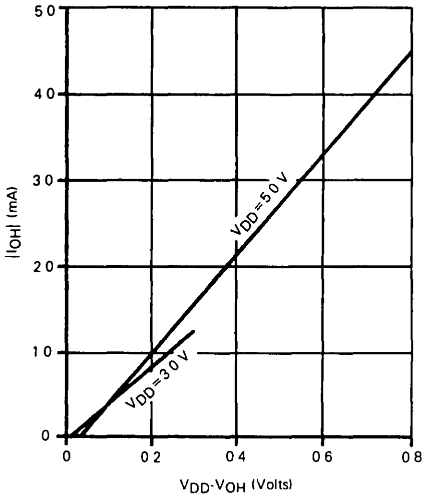


Figure 23. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

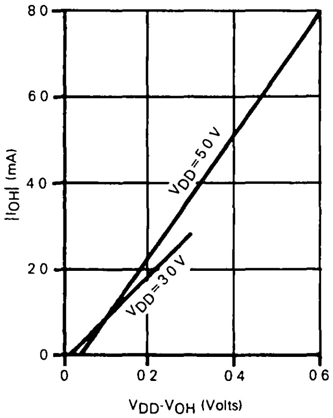


Figure 24. Typical V_{OH} vs I_{OH} for PD1-PD4

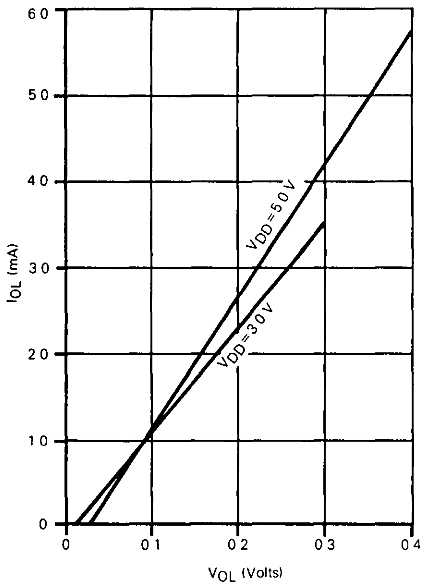


Figure 25. Typical V_{OL} vs I_{OL} for All Ports

CONTROL TIMING

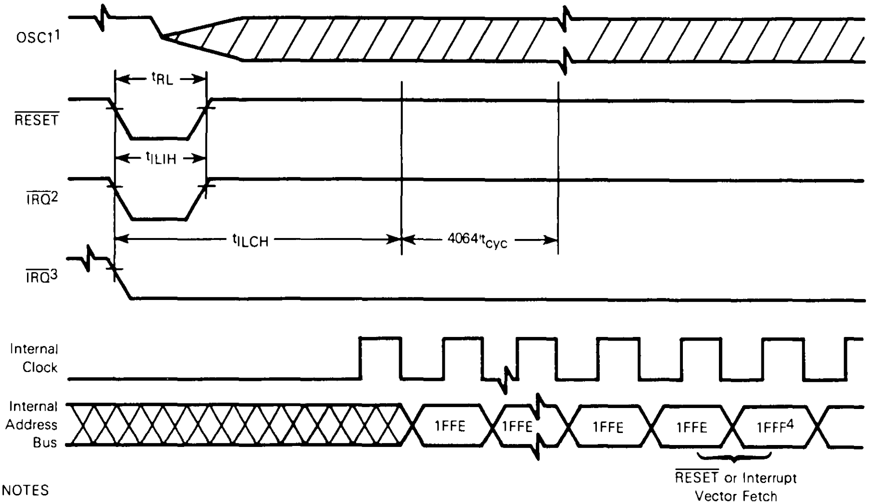
($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc}-2$)		—	2.1	
External Clock ($f_{osc}-2$)		dc	2.1	
Cycle Time (see Figure 29)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 29)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 29)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 27)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 27)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.


OSC1¹
t_{RL}
$\overline{\text{RESET}}$
t_{ILIH}
$\overline{\text{IRQ}}$²
t_{ILCH}
4064 t_{cyc}
$\overline{\text{IRQ}}$³
Internal Clock
Internal Address Bus
1FFE
1FFE
1FFE
1FFE
1FFF⁴
$\overline{\text{RESET}}$ or Interrupt Vector Fetch

NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3 $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option
4 $\overline{\text{RESET}}$ vector address shown for timing example

Figure 26. Stop Recovery Timing Diagram

CONTROL TIMING

($V_{DD} = 3.3$ Vdc ± 0.3 Vdc, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 2.0 2.0	MHz
Internal Operating Frequency Crystal ($f_{osc} - 2$) External Clock ($f_{osc} - 2$)	f_{op}	 — dc	 1.0 1.0	MHz
Cycle Time (see Figure 29)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 29)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 26)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 29)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 27) Input Capture Pulse Period (see Figure 27)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

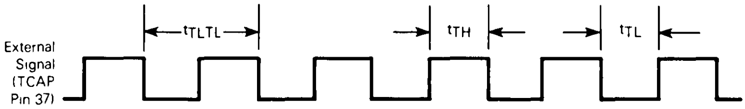


Figure 27. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 240	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

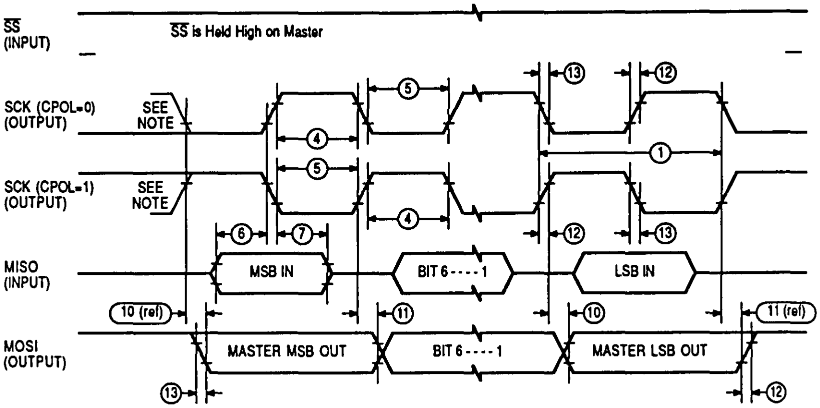
SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H) (see Figure 28)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 1.0	 — —	 t_{cyc} µs
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 500	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 500	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 720 400	 — —	 µs ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 720 400	 — —	 µs ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 200 200	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 200 200	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 250	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 500	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 500	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 200 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 200 2.0	 ns µs

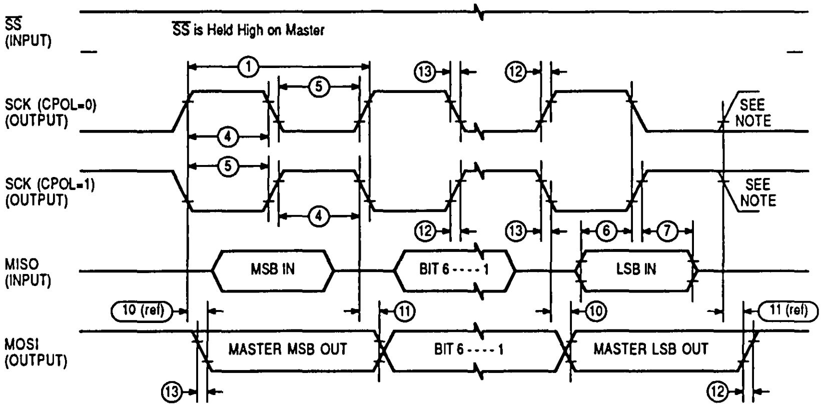
*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MOSI (OUTPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

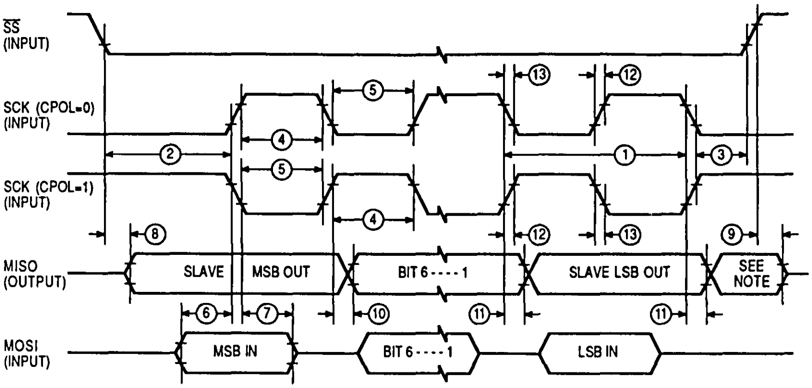
a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MOSI (OUTPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

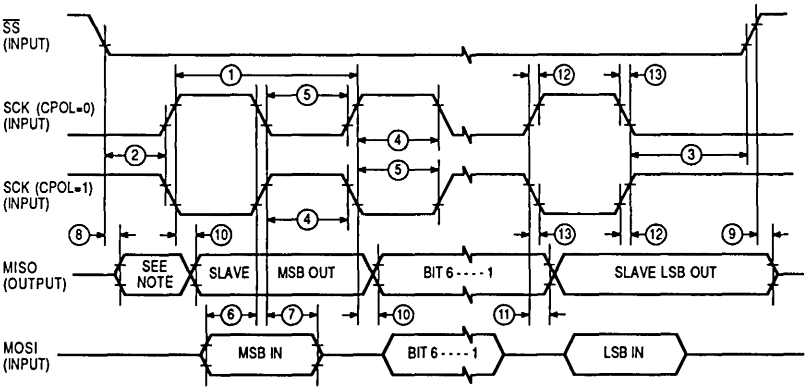
b) SPI MASTER TIMING (CPHA=1)

Figure 28. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally MSB of character just received.

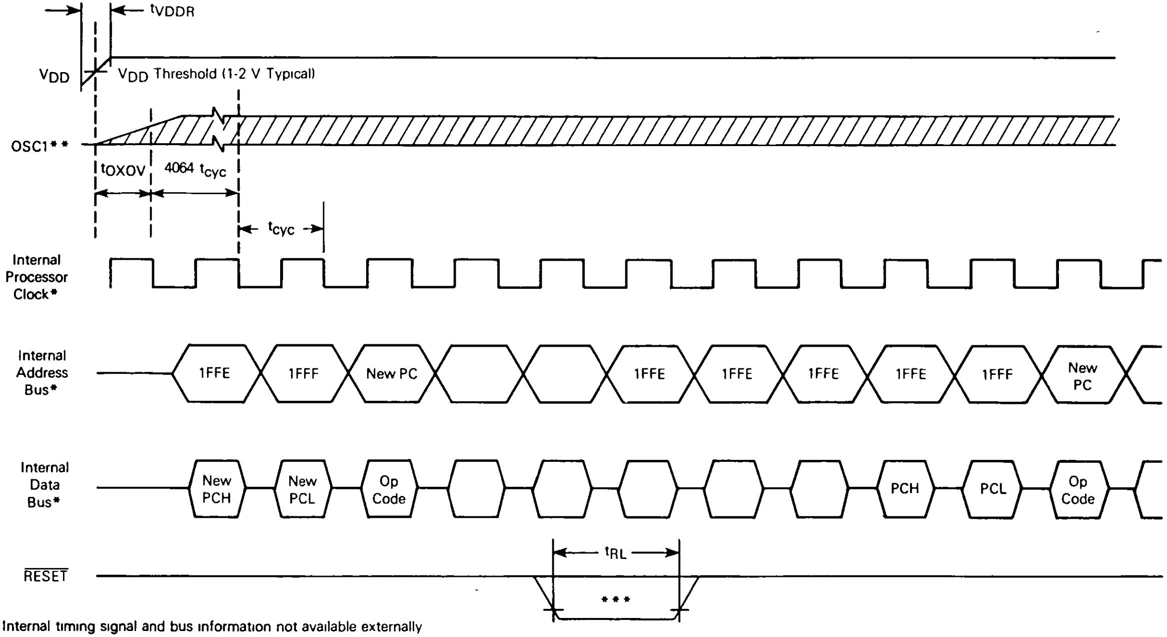
c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE: Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA=1)

Figure 28. SPI Timing Diagrams (Sheet 2 of 2)


tVDDR
VDD
VDD Threshold (1-2 V Typical)
OSC1**
tOXOV
4064 tcyc
tcyc
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
tRL
RESET
***

*Internal timing signal and bus information not available externally
**OSC1 line is not meant to represent frequency It is only used to represent time
***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 29. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC705C8 device.

OTPROM MCU Devices

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C −40°C to +85°C	MC68HC705C8P MC68HC705C8CP
PLCC (FN Suffix)	0°C to +70°C −40°C to +85°C	MC68HC705C8FN MC68HC705C8CFN

EPROM MCU Device

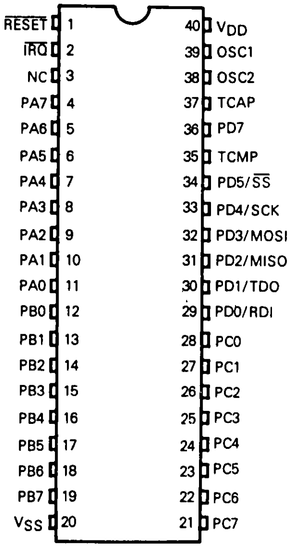
Package Type	Temperature	MC Part Number
Cerdip (S Suffix)	0°C to +70°C −40°C to +85°C	MC68HC705C8S MC68HC705C8CS

MECHANICAL DATA

This section contains the pin assignments and package dimensions for the MC68HC705C8.

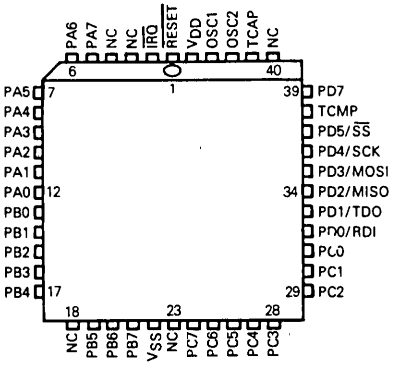
PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE



RESET 1
IRQ 2
NC 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE



PA6
PA7
NC
NC
IRQ
RESET
VDD
OSC1
OSC2
TCAP
NC
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
VSS
NC
PC7
PC6
PC5
PC4
PC3

NOTE: Bulk substrate tied to V_{SS}.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC805B6

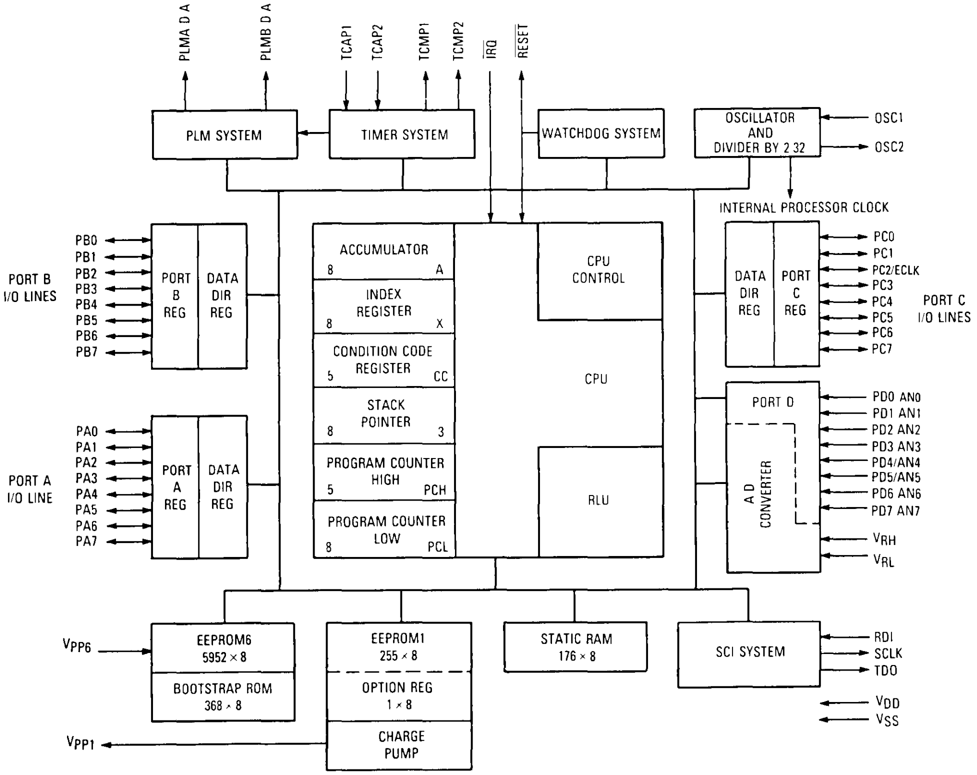
Technical Summary
8-Bit Microcontroller Unit

The MC68HC805B6 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are shown below and at the top of page 2.

- On-Chip Oscillator with Crystal/Ceramic Resonator
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 256 Bytes of Byte-Erasable EEPROM (EEPROM1)

BLOCK DIAGRAM


PLMA D A
PLMB D A
TCAP1
TCAP2
TCMP1
TCMP2
IRQ
RESET
PLM SYSTEM
TIMER SYSTEM
WATCHDOG SYSTEM
OSCILLATOR AND DIVIDER BY 2 32
OSC1
OSC2
INTERNAL PROCESSOR CLOCK
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PORT B I/O LINES
PORT B REG
DATA DIR REG
ACCUMULATOR 8 A
INDEX REGISTER 8 X
CONDITION CODE REGISTER 5 CC
STACK POINTER 8 3
PROGRAM COUNTER HIGH 5 PCH
PROGRAM COUNTER LOW 8 PCL
CPU CONTROL
CPU
ALU
DATA DIR REG
PORT C REG
PC0
PC1
PC2/ECLK
PC3
PC4
PC5
PC6
PC7
PORT C I/O LINES
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A I/O LINE
PORT A REG
DATA DIR REG
PORT D
A D CONVERTER
PD0 AN0
PD1 AN1
PD2 AN2
PD3 AN3
PD4/AN4
PD5/AN5
PD6 AN6
PD7 AN7
VRH
VRL
VPP6
EEPROM6 5952 × 8
BOOTSTRAP ROM 368 × 8
EEPROM1 255 × 8
OPTION REG 1 × 8
CHARGE PUMP
STATIC RAM 176 × 8
SCI SYSTEM
RDI
SCLK
TDO
VDD
VSS
VPP1

This document contains information on a new product Specifications and information herein are subject to change without notice.

FEATURES (Continued)

- 368 Bytes of Bootstrap ROM
- 5952 Bytes of Bulk-Erasable EEPROM (EEPROM6)
- 24 Bidirectional I/O Lines and 8 Input-Only Lines
- Serial Communications Interface (SCI) System
- 8-Channel A/D Converter
- Watchdog System
- Bootstrap Capability
- Power-Saving STOP and WAIT Modes
- Single 3.0- to 6.0-Volt Supply
- Fully Static Operation
- Two Pulse-Length Modulation Systems (D/A)
- 16-Bit Timer with Two Input Capture and Two Output Compare Functions
- 2-Channel Pulse Length Modulator
- Slow Mode Option Divides the Basic Clock Frequency by 16

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{\text{IRQ}}$

This pin is a programmable option that provides four different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail. This pin is also used to change operating modes. See Table 2.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, or an external signal connects to these pins providing a system clock. The oscillator frequency is two times the internal bus rate (or 32 times as a software option).

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(d).

INPUT CAPTURE (TCAP1)

This pin controls the input capture 1 feature for the on-chip programmable timer. This pin is also used to change operating modes. See Table 2.

INPUT CAPTURE (TCAP2)

This pin controls the input capture 2 feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP1)

This pin provides an output for the output compare 1 feature of the on-chip timer.

OUTPUT COMPARE (TCMP2)

This pin provides an output for the output compare 2 feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low. The voltage on this pin affects the mode of operation (see Table 2).

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0/AN0–PD7/AN7)

These eight lines comprise port D, a fixed input port. Enabling the A/D function affects this port. Port D accepts the eight analog inputs when the A/D is enabled. Port D can be used for digital input during a conversion sequence, but this may inject noise on the analog signals,

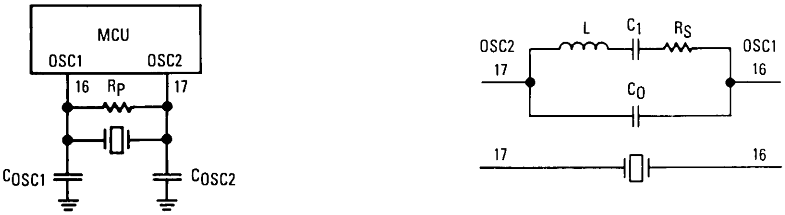
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

Ceramic Resonator

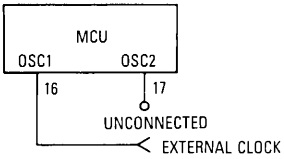
	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4, 3	μF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	—

(a) Crystal/Ceramic Resonator Parameters



(b) Crystal/Ceramic Resonator Oscillator Connections

(c) Equivalent Crystal Circuit



(d) External Clock Source Connections

Figure 1. Oscillator Connections

reducing the conversion accuracy. Also, a digital read of port D with levels other than V_{DD} or V_{SS} on the pins results in greater power dissipation during the read cycle. Refer to **PROGRAMMING** for additional information.

PLMA

This pin is the output of the pulse-length modulation converter A. See **PULSE-LENGTH D/A CONVERTERS** for further information.

PLMB

This pin is the output of the pulse-length modulation converter B. See **PULSE-LENGTH D/A CONVERTERS** for further information.

RDI (Receive Data In)

This pin is the input of the SCI. See **SERIAL COMMUNICATIONS INTERFACE** for more information

TDO (Transmit Data Out)

This pin is the output of the serial communications transmitter. See **SERIAL COMMUNICATIONS INTERFACE** for more information.

SCLK

This pin is the clock output pin of the SCI transmitter. See **SERIAL COMMUNICATIONS INTERFACE** for more information.

V_{PP1}

This pin is the EEPROM1 programming voltage output. See **EEPROM** for further information.

V_{PP6}

This pin is the EEPROM6 programming voltage. See **EEPROM** for further information.

V_{RH}

This pin is the positive reference voltage for the A/D converter.

V_{RL}

This pin is the negative reference voltage for the A/D converter.

INPUT/OUTPUT PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 2 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Under software control, the PC2 pin can become the CPU clock output. If this option is selected, the corresponding DDR bit is automatically set, and bit 2 of port C always reads the output data latch. The other port C pins are not affected by this feature. See ECLK bit in **EEPROM/ECLK Control Register** for details.

Table 1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port that monitors the external pins whenever the A/D is disabled. After reset, all eight bits become digital inputs because all special function drivers are disabled. Port D is always a digital input, whether the A/D is on or off.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI) PROGRAMMING

The SCI uses two or three pins for its functions: RDI for its receive data input, TDO for its transmit data output, and SCLK to output the transmitter clock, if needed.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 3. The locations consist of user ROM, user RAM, EEPROM, bootstrap ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF0 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

EEPROM

The MCU has 256 bytes of byte-erasable EEPROM, called EEPROM1, and 5952 bytes of bulk-erasable EEPROM,

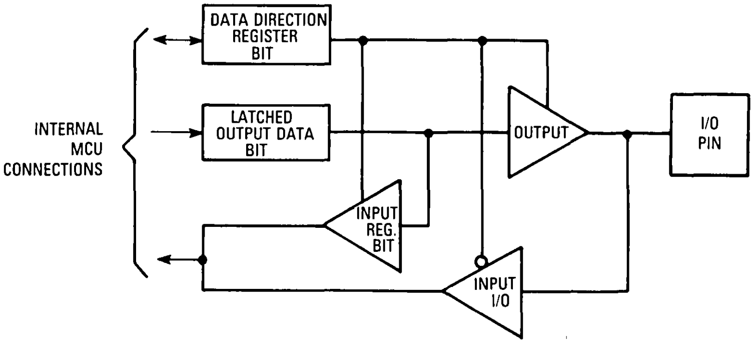


Figure 2. Typical Port I/O Circuit

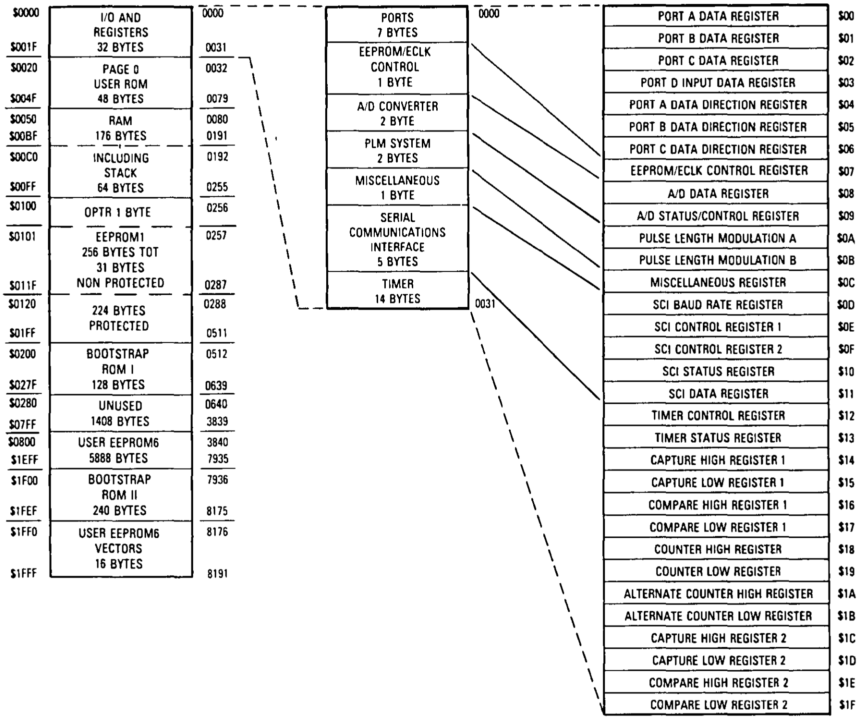

$0000
I/O AND REGISTERS 32 BYTES
$001F
$0020
PAGE 0 USER ROM 48 BYTES
$004F
$0050
RAM 176 BYTES
$00BF
$00C0
INCLUDING STACK 64 BYTES
$00FF
$0100
OPTR 1 BYTE
$0101
EEPROM1 256 BYTES TOT 31 BYTES NON PROTECTED
$011F
$0120
224 BYTES PROTECTED
$01FF
$0200
BOOTSTRAP ROM I 128 BYTES
$027F
$0280
UNUSED 1408 BYTES
$07FF
$0800
USER EEPROM6 5888 BYTES
$1EFF
$1F00
BOOTSTRAP ROM II 240 BYTES
$1FEF
$1FF0
USER EEPROM6 VECTORS 16 BYTES
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
0257
0287
0288
0511
0512
0639
0640
3839
3840
7935
7936
8175
8176
8191
PORTS 7 BYTES
EEPROM/ECLK CONTROL 1 BYTE
A/D CONVERTER 2 BYTE
PLM SYSTEM 2 BYTES
MISCELLANEOUS 1 BYTE
SERIAL COMMUNICATIONS INTERFACE 5 BYTES
TIMER 14 BYTES
0000
0031
PORT A DATA REGISTER $00
PORT B DATA REGISTER $01
PORT C DATA REGISTER $02
PORT D INPUT DATA REGISTER $03
PORT A DATA DIRECTION REGISTER $04
PORT B DATA DIRECTION REGISTER $05
PORT C DATA DIRECTION REGISTER $06
EEPROM/ECLK CONTROL REGISTER $07
A/D DATA REGISTER $08
A/D STATUS/CONTROL REGISTER $09
PULSE LENGTH MODULATION A $0A
PULSE LENGTH MODULATION B $0B
MISCELLANEOUS REGISTER $0C
SCI BAUD RATE REGISTER $0D
SCI CONTROL REGISTER 1 $0E
SCI CONTROL REGISTER 2 $0F
SCI STATUS REGISTER $10
SCI DATA REGISTER $11
TIMER CONTROL REGISTER $12
TIMER STATUS REGISTER $13
CAPTURE HIGH REGISTER 1 $14
CAPTURE LOW REGISTER 1 $15
COMPARE HIGH REGISTER 1 $16
COMPARE LOW REGISTER 1 $17
COUNTER HIGH REGISTER $18
COUNTER LOW REGISTER $19
ALTERNATE COUNTER HIGH REGISTER $1A
ALTERNATE COUNTER LOW REGISTER $1B
CAPTURE HIGH REGISTER 2 $1C
CAPTURE LOW REGISTER 2 $1D
COMPARE HIGH REGISTER 2 $1E
COMPARE LOW REGISTER 2 $1F

Figure 3. Memory Map

called EEPROM6. An internal charge pump, connected to the V_{PP1} pin, supplies a high voltage for erase and programming of EEPROM1. The V_{PP1} pin should be left open. For erase and programming of EEPROM6, a high voltage must be applied to the V_{PP6} pin.

CAUTION

When programming voltage is not required at V_{PP6}, a voltage equal to V_{DD} must be applied at all times, including power-on. Applying a voltage lower than V_{DD} could damage the device. An external high voltage should **not** be applied to the V_{PP1} pin.

To provide a higher degree of security for stored data, there is no bulk or row erase in EEPROM1.

Read Operation

To read data from EEPROM1, the E1LAT bit must be zero. When E1LAT is zero, the E1PGM and E1ERA bits are forced to zero, and EEPROM1 is read as if it were a normal ROM. The V_{PP1} charge pump generator is off since E1PGM is zero. If a read is performed while E1LAT is set, data will be read as $FF.

NOTE

When not performing a programming or erase operation on the EEPROM, remain in read mode (E1LAT = 0).

Erase Operation

The bootstrap loading program contains a routine that will erase and program both EEPROM1 and EEPROM6. See **Bootstrap Mode** for more information.

To erase a byte of EEPROM1, set E1LAT and E1ERA to one, write to the address to be erased, and set E1PGM for a time t_{ERAL}. After the required erase time, E1LAT must be cleared, which resets E1ERA and E1PGM. To erase a second word, E1LAT must be cleared before it is set, or the erase will have no effect. This procedure is done to increase the security of the stored data. While an erase is being performed, any access to the EEPROM1 will return $FF. Data written in an erase operation is not used; therefore, its value is not significant. User programs must be running from ROM or RAM, since EEPROM1 has its address and data buses latched.

The EEPROM6 can only be bulk erased in the bootstrap mode (see Table 2).

Programming Operation

To program a byte of EEPROM1, set the E1LAT bit, write data to the desired address, and set the E1PGM bit. After the required programming delay (t_{PROG1}), E1LAT must be cleared, which also resets E1PGM. During a programming operation, any access to EEPROM1 will return $FF.

NOTE

> To program a byte correctly, the byte must have been previously erased.

To program a second word, E1LAT must be cleared before it is set, or the programming will have no effect. This procedure is done to increase the security of the stored data. User programs must be running from RAM or ROM since EEPROM1 will have its data buses latched.

EEPROM6 can only be programmed in bootstrap mode. User programs must be running from RAM or ROM since EEPROM6 will have its data buses latched during the programming operation.

Bootstrap Mode

The MCU contains a bootstrap program in ROM that can be used to program the EEPROMs. Table 2 shows how the operating modes are selected.

Table 2. Operating Mode Selection

$\overline{\text{RESET}}$ Pin	$\overline{\text{IRQ}}$ Pin	TCAP1 Pin	Mode
_/‾	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Normal
_/‾	+9 Volts	V_{DD}	Bootstrap
V_{SS}	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Reset Condition

The hold time on the $\overline{\text{IRQ}}$ and TCAP1 pins is two clock cycles after $\overline{\text{RESET}}$ goes high.

Four bootstrap modes are available, selected by the state of pins PD3/AN3 and PD4/AN4 when bootstrap mode is selected. Figure 4 is a flowchart of the operating modes of the MCU. The four modes available are 1) EEPROM parallel bootstrap, 2) EEPROM serial bootstrap, 3) Serial operation, and 4) RAM bootstrap.

EEPROM Parallel Bootstrap

The EEPROM parallel bootstrap is entered by setting the bits as follows:

PD3/AN3 = 0
PD4/AN4 = 0

Figure 5 shows a schematic diagram of a circuit that can be used for the parallel bootstrap operation. First, the EEPROM6 is erased (if V_{PP6} is applied) and checked for complete erasure. If a byte other than $FF is detected, the red LED is turned on, and the erase is repeated until erasure is complete. Before erasing EEPROM1, the red LED is turned off. EEPROM1 is erased and checked for complete erasure (repeating the erase if needed), and the security bit is cleared.

After both EEPROMs are erased, they are loaded in increasing address order. Segments containing no EEPROM are skipped by the loader. An algorithm in the bootstrap program skips $FF data bytes to speed the loading process. Parallel data is entered through port A; the 13-bit address is output on port B and port C, C0 to C4. If data comes from an external EEPROM, the handshake can be disabled by connecting C5 and C6. If data is supplied by a parallel interface, handshake is supplied by C5 and C6, with the timing shown in Figure 6.

Both EEPROM1 and EEPROM6 can be loaded from an external source. If one segment, such as EEPROM1, is to retain unprogrammed (previously erased) data, then the data for this segment should be all FFs.

EEPROM Serial Bootstrap

The EEPROM serial bootstrap is entered by setting the bits as follows:

PD3/AN3 = 1
PD4/AN4 = 0

Figure 7 shows a schematic diagram of a circuit that can be used for the serial bootstrap operation. First, the EEPROM6 is erased (if V_{PP6} is applied) and checked for complete erasure. If a byte other than $FF is detected, the red LED is turned on, and the erase is repeated until erasure is completed. Before erasing EEPROM1, the red LED is turned off, EEPROM1 is erased and checked for complete erasure (repeating the erase if needed), and the security bit is cleared.

The serial routine communicates through the SCI with an external host computer such as an IBM™ PC using an RS-232C link with the following parameters:

9600 Baud
8 Bit
No Parity
Full Duplex

The data format is 8-bit binary, so a complementary program in the host is required to supply the necessary format. This program is available from Motorola for the IBM PC. The format required for the serial loader is as follows:

(address high) (address low) (data)

The protocol for host communication is as follows:

1. The MCU sends the last byte programmed to the host to allow the host to verify correct programming (data is undetermined for the first byte).
2. The host sends the three data bytes to the MCU.
3. Non-$FF data is programmed at the address provided in port, RAM, or EEPROM. For $FF data, no programming takes place, and data in the accessed location is returned as a prompt.
4. Loop to one.

Addresses from $50 to $82 are used by the loader, and are not available during serial loading. All addresses between $83 and $E4 are available for user test programs

---

IBM is a trademark of International Business Machines

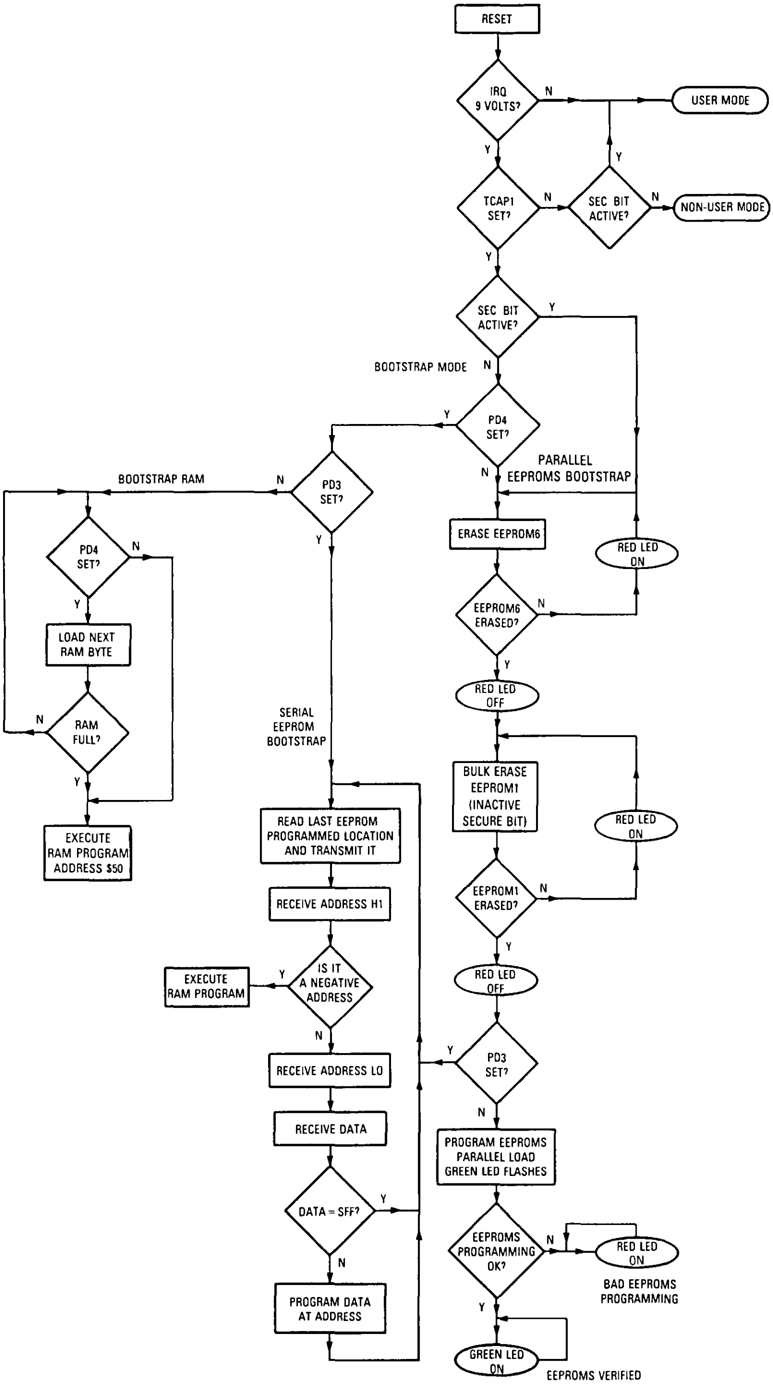

RESET
IRQ 9 VOLTS?
N
USER MODE
Y
TCAP1 SET?
N
SEC BIT ACTIVE?
Y
N
NON-USER MODE
Y
SEC BIT ACTIVE?
Y
BOOTSTRAP MODE
N
PD4 SET?
Y
N
PARALLEL EEPROMS BOOTSTRAP
BOOTSTRAP RAM
N
PD3 SET?
Y
PD4 SET?
N
Y
LOAD NEXT RAM BYTE
N
RAM FULL?
Y
EXECUTE RAM PROGRAM ADDRESS $50
SERIAL EEPROM BOOTSTRAP
READ LAST EEPROM PROGRAMMED LOCATION AND TRANSMIT IT
RECEIVE ADDRESS HI
IS IT A NEGATIVE ADDRESS
Y
EXECUTE RAM PROGRAM
N
RECEIVE ADDRESS LO
RECEIVE DATA
DATA = $FF?
Y
N
PROGRAM DATA AT ADDRESS
ERASE EEPROM6
RED LED ON
EEPROM6 ERASED?
N
Y
RED LED OFF
BULK ERASE EEPROM1 (INACTIVE SECURE BIT)
RED LED ON
EEPROM1 ERASED?
N
Y
RED LED OFF
PD3 SET?
Y
N
PROGRAM EEPROMS PARALLEL LOAD GREEN LED FLASHES
EEPROMS PROGRAMMING OK?
N
RED LED ON
BAD EEPROMS PROGRAMMING
Y
GREEN LED ON
EEPROMS VERIFIED

Figure 4. Operating Modes Flowchart

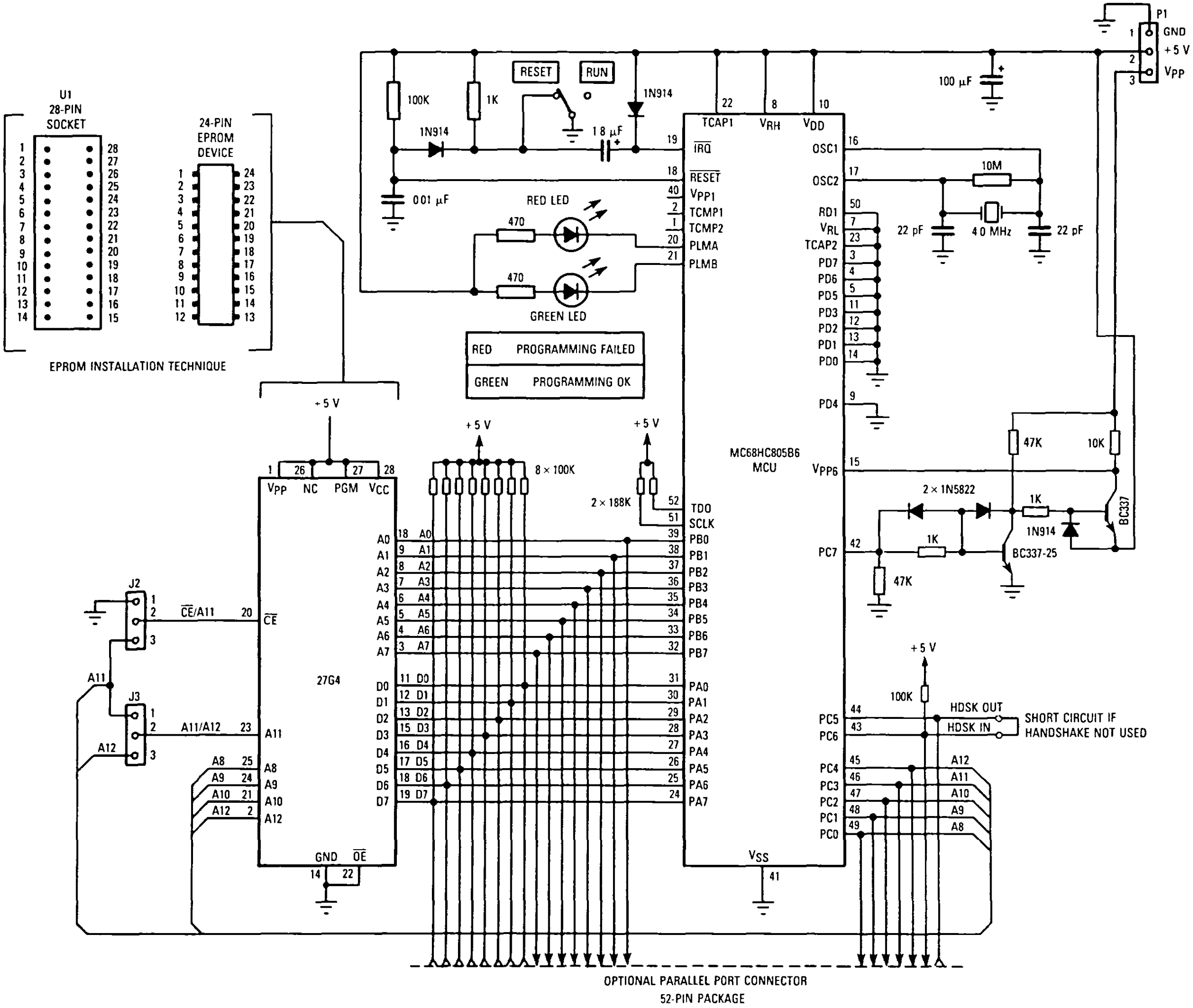

U1
28-PIN
SOCKET
24-PIN
EPROM
DEVICE
EPROM INSTALLATION TECHNIQUE
100K
1K
RESET
RUN
1N914
1N914
.001 µF
1.8 µF
RED LED
GREEN LED
470
470
RED PROGRAMMING FAILED
GREEN PROGRAMMING OK
TCAP1
VRH
VDD
IRQ
RESET
VPP1
TCMP1
TCMP2
PLMA
PLMB
OSC1
OSC2
RD1
VRL
TCAP2
PD7
PD6
PD5
PD3
PD2
PD1
PD0
PD4
VPP6
PC7
MC68HC805B6
MCU
TDO
SCLK
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PC5
PC6
PC4
PC3
PC2
PC1
PC0
VSS
10M
22 pF
4.0 MHz
22 pF
100 µF
P1
GND
+5 V
VPP
47K
10K
2 × 1N5822
1K
1N914
BC337
1K
BC337-25
47K
+5 V
100K
HDSK OUT
HDSK IN
SHORT CIRCUIT IF
HANDSHAKE NOT USED
A12
A11
A10
A9
A8
+5 V
8 × 100K
+5 V
2 × 188K
27G4
VPP
NC
PGM
VCC
A0
A1
A2
A3
A4
A5
A6
A7
D0
D1
D2
D3
D4
D5
D6
D7
CE
A11
A8
A9
A10
A12
GND
OE
J2
J3
CE/A11
A11/A12
A11
A12
OPTIONAL PARALLEL PORT CONNECTOR
52-PIN PACKAGE

Figure 5. EEPROM Parallel Bootstrap Schematic Diagram

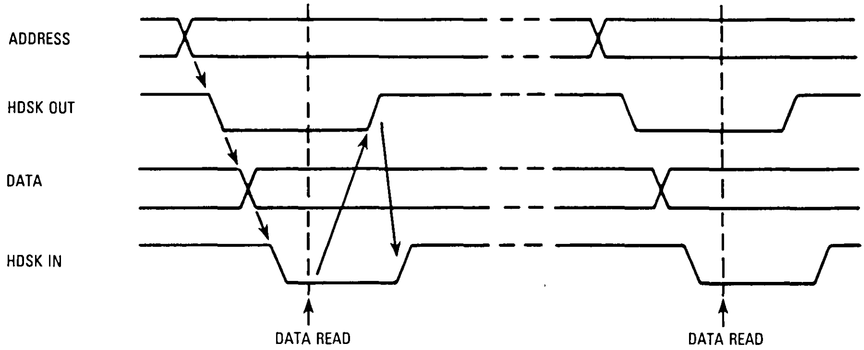


Figure 6. EEPROM Load Timing Diagram

in RAM. Addresses $E4 and above are pointed to by vectors (see Table 3), making vectors (except reset) available to the user. A 10-byte stack is reserved at the top of the RAM to allow, for example, one interrupt and two subroutine levels.

Table 3. Bootstrap Vector Targets in RAM

Vector Targets	Vector Address
SCI Interrupt	$E4
Timer Overflow	$E7
Timer Output Compare	$EA
Timer Input Capture	$ED
External Interrupt	$F0
Software Interrupt	$F3

Program execution is started by sending a negative address (bit 7 set); execution begins at address $83.

Serial Operation

Serial operations other than serial bootstrap are entered by setting the bits as follows:

PD3/AN3 = 1
PD4/AN4 = 1

If the security bit is active (0), the program performs an erase operation. First, the EEPROM6 is erased (if V_{PP6} is applied) and checked for complete erasure. If a byte other than $FF is detected, the red LED is turned on, and the erase is repeated until erasure is complete. Before erasing EEPROM1, the red LED is turned off, EEPROM1 is erased and checked for complete erasure (repeating the erase if needed), and the security bit is cleared.

The serial protocol and format are identical to those described in **EEPROM Serial Bootstrap**, except that here the serial routines are entered without going through an erase cycle if the SEC bit is not active.

Serial read can be performed by sending the address desired, followed by a data byte of $FF. The $FF byte causes a skip of the programming operation, and the byte returned is the actual content of the addressed location.

RAM Parallel Bootstrap

The RAM parallel bootstrap is entered by setting the bits as follows:

PD3/AN3 = 0
PD4/AN4 = 1

If the security bit is active (0), the program performs an erase operation. First, the EEPROM6 is erased (if V_{PP6} is applied) and checked for complete erasure. If a byte other than $FF is detected, the red LED is turned on, and the erase is repeated until erasure is complete. Before erasing EEPROM1, the red LED is turned off, EEPROM1 is erased and checked for complete erasure (repeating the erase if needed), and the security bit is cleared.

If the security bit is not active (1), the RAM bootstrap program is executed and loads the RAM with external data. When the RAM gets full, control passes to the loaded program at address $50. Refer to Figure 8 for details of the external connections. Up to eight programs can be loaded from the EPROM. Selection is made using the switches connected to the higher order address bits of the EPROM.

No handshake is provided in this mode. Address lines C0 to C4 and C7 will be zero; C5 will be high; and C6 is unused. If a user program sets C0 to a logic one, the EPROM is disabled, and ports A and B are available.

EEPROM/ECLK Control Register $07

	7	6	5	4	3	2	1	0
	—	—	—	—	ECLK	E1ERA	E1LAT	E1PGM
RESET	0	0	0	0	0	0	0	0

Bit 7–4 are reserved for factory test.

ECLK — ECLK Control

1 = I/O port function of PC2 is forced to output mode, and PC2 outputs the ECLK CPU clock.

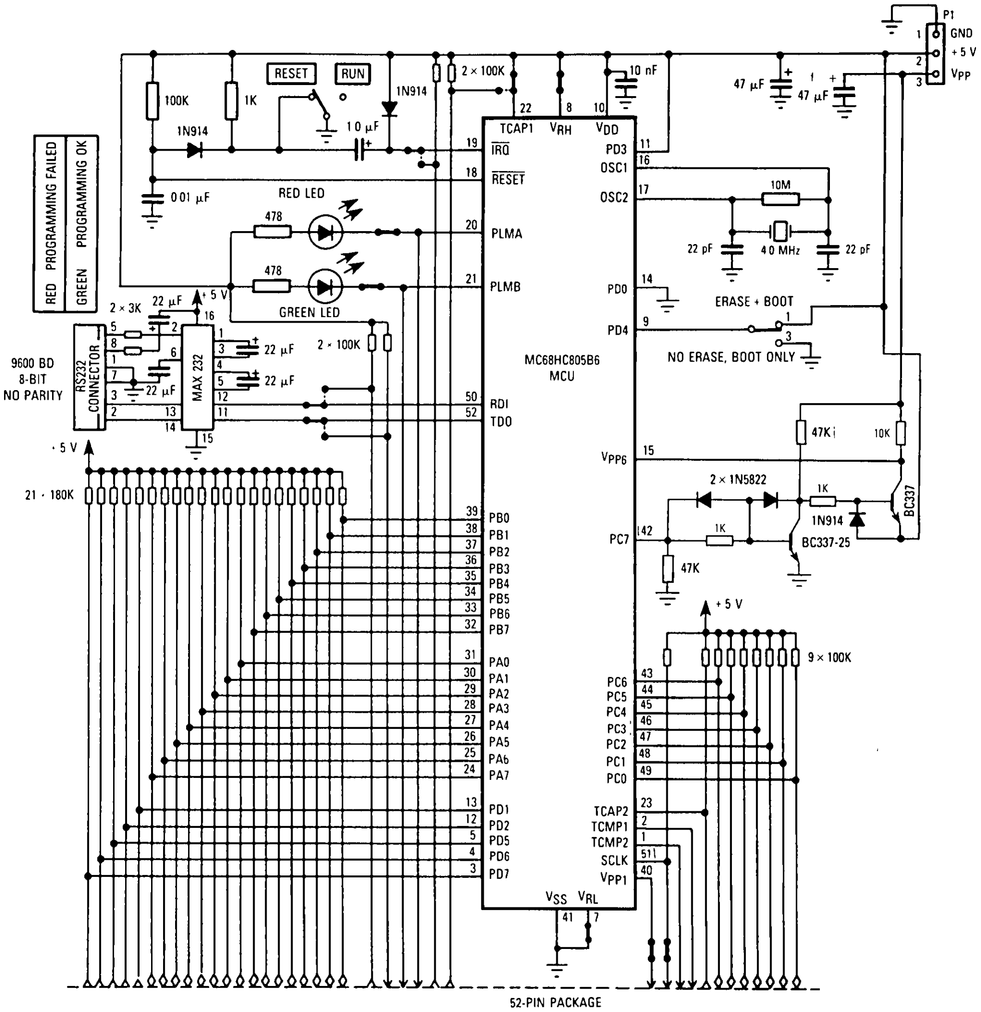

P1
GND
+5 V
VPP
1
2
3
47 µF
47 µF
10 nF
2 × 100K
RESET
RUN
1N914
100K
1K
1N914
1.0 µF
0.01 µF
RED LED
478
478
GREEN LED
RED PROGRAMMING FAILED
GREEN PROGRAMMING OK
2 × 3K
22 µF
+5 V
16
RS232 CONNECTOR
MAX 232
9600 BD
8-BIT
NO PARITY
22 µF
22 µF
22 µF
2 × 100K
+5 V
21 · 180K
TCAP1
VRH
VDD
IRQ
RESET
PLMA
PLMB
RDI
TDO
PD3
OSC1
OSC2
PD0
PD4
10M
22 pF
4.0 MHz
22 pF
ERASE + BOOT
NO ERASE, BOOT ONLY
MC68HC805B6
MCU
47K
10K
VPP6
2 × 1N5822
1K
1N914
BC337
1K
BC337-25
PC7
47K
+5 V
9 × 100K
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PD1
PD2
PD5
PD6
PD7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
TCAP2
TCMP1
TCMP2
SCLK
VPP1
VSS
VRL
52-PIN PACKAGE

Figure 7. RAM/EEPROM Serial Bootstrap Schematic Diagram

0 = PC2 functions as a regular I/O pin.

When ECLK is set, the corresponding DDR bit is set, and port C, bit 2 always reads the output data latch. Other port C pins are not affected.

E1ERA — EEPROM1 Erase

1 = An erase will take place if E1LAT and E1PGM are both one.

0 = A programming operation will take place if E1LAT and E1PGM are both one.

If E1LAT = 0, E1ERA is held at zero. Once an EEPROM address is selected, E1ERA cannot be changed.

E1LAT — EEPROM1 Latch Enable

1 = Address and data can be latched into the EEPROM for programming or erase operation if E1PGM = 0.

0 = Data can be read from the EEPROM, and the E1ERA and E1PGM bits are cleared.

After the programming or erase time, the E1LAT bit must be reset in order to reset the E1ERA and E1PGM bits.

E1PGM — EEPROM1 Program Mode

1 = Charge pump generator is on, and the resulting high voltage is applied to the EEPROM1 array.

0 = Charge pump generator is off.

E1PGM cannot be set before the data is selected; it can only be reset by resetting E1LAT.

The V_{PP1} charge pump is not affected by the WAIT mode; therefore, WAIT can be used for the erase or programming delay time. If STOP mode is entered, the EEPROM is set to read mode.

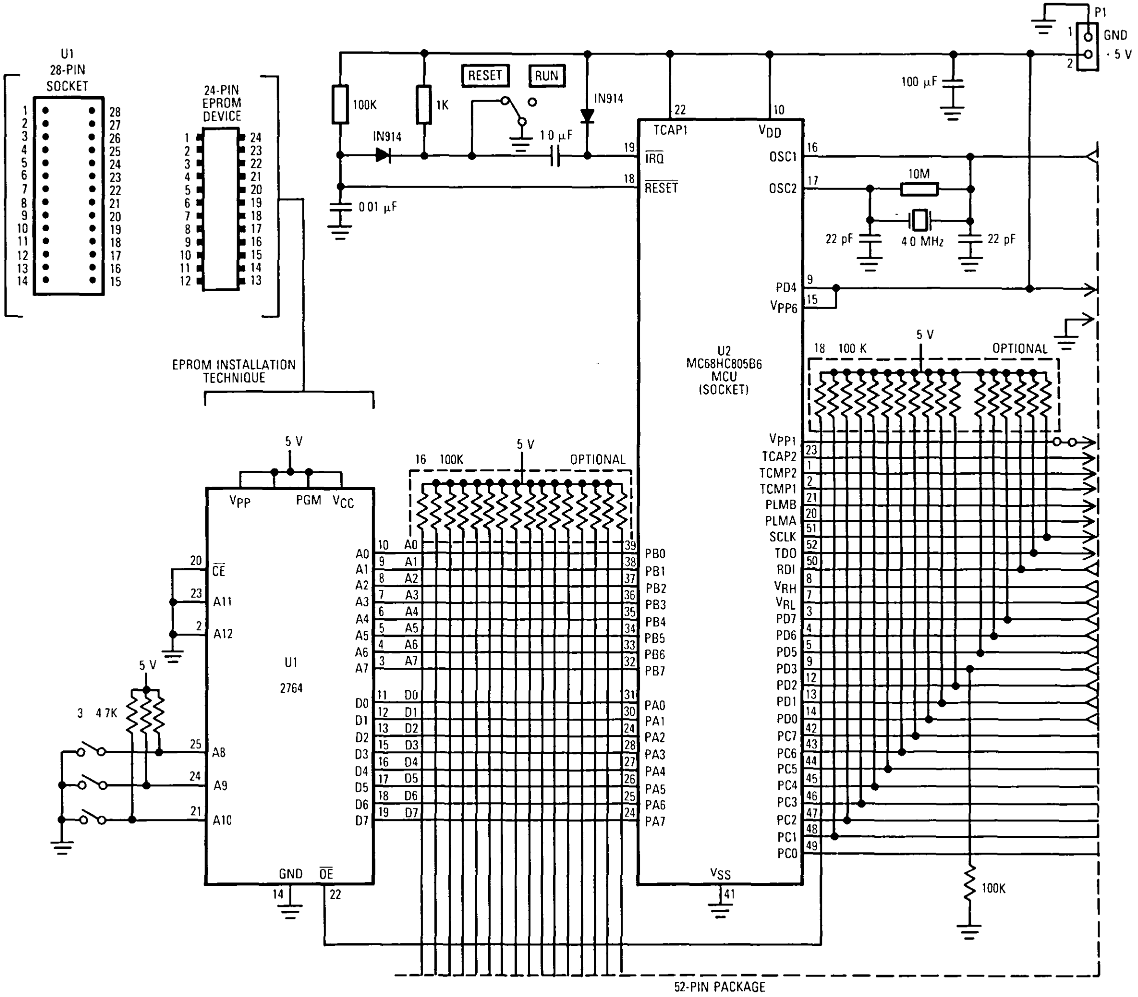

U1
28-PIN
SOCKET
24-PIN
EPROM
DEVICE
EPROM INSTALLATION
TECHNIQUE
100K
1K
IN914
RESET
RUN
IN914
10 μF
0.01 μF
P1
GND
+5 V
100 μF
22
10
TCAP1
VDD
19
IRQ
18
RESET
16
OSC1
17
OSC2
10M
22 pF
4.0 MHz
22 pF
PD4
VPP6
U2
MC68HC805B6
MCU
(SOCKET)
18 100 K
5 V
OPTIONAL
VPP1
TCAP2
TCMP2
TCMP1
PLMB
PLMA
SCLK
TDO
RDI
VRH
VRL
PD7
PD6
PD5
PD3
PD2
PD1
PD0
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
VSS
41
100K
16 100K
5 V
OPTIONAL
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
5 V
VPP
PGM
VCC
CE
A11
A12
A8
A9
A10
U1
2764
A0
A1
A2
A3
A4
A5
A6
A7
D0
D1
D2
D3
D4
D5
D6
D7
GND
OE
14
22
5 V
3 4.7K
52-PIN PACKAGE

Figure 8. RAM Parallel Bootstrap Schematic Diagram

The V_{PP1} charge pump generator is normally supplied by the CPU clock, but for very low clocking frequencies, the A/D RC oscillator should be used. See **A/D CONVERTER** for more information.

Options Register (OPTR) $0100

7	6	5	4	3	2	1	0
—	—	—	—	—	—	EE1P	SEC
RESET.							
U	U	U	U	U	U	U	U

(U = Unaffected)

EE1P — EEPROM1 Part 2 Protect
- 1 = EEPROM1 not protected.
- 0 = EEPROM1 addresses from $0120 to $01FF are read only, and attempts to erase or program this area will be unsuccessful.

When this bit is erased to one, protection remains until the next external or power-up reset occurs.

SEC — EEPROM1 Security Bit
- 1 = Security not active.
- 0 = EEPROM1 contents secured from external access.

The SEC bit can only be changed to one by entering EEPROM bootstrap mode, which erases the entire EEPROM. When SEC is changed, the new value has no effect until the next external or power-up reset.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

7	0
A	

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.

7	0
X	

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed.

12	0
PC	

STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

12					7		0
0	0	0	0	0	1	1	SP

CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

RESETS

The MCU can be reset two ways: by initial power-up (POR) and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON RESET

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. Once the internal processor becomes active, a delay of 4064 clock cycles (t_{PORL}) occurs.If the $\overline{\text{RESET}}$ pin is low at the end of t_{PORL}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

Miscellaneous Register ($0C)

	7	6	5	4	3	2	1	0
	POR	INTP	INTN	INTE	SFA	SFB	SM	WDOG
RESET	U	0	0	1	0	0	0	0

(U = Unaffected)

POR — Power-On Reset
- 1 = The reset occurring is a power-on, not external, reset
- 0 = Power-on reset not in progress

INTP — External Interrupt Positive
- Allows a choice of $\overline{\text{IRQ}}$ sensitivity, with INTN. See Table 4.

INTN — External Interrupt Negative
- Allows a choice of $\overline{\text{IRQ}}$ sensitivity, with INTP. See Table 4.

INTE — External Interrupt Enable
- Allows the user to enable or disable the external interrupt function.

SFA — Slow/Fast Selection for PLMA
- 1 = Slow speed used for PLMA (4096 times the timer clock period).
- 0 = Fast speed used for PLMA (256 times the timer clock period). See **PULSE-LENGTH D/A CONVERTERS.**

SFB — Slow/Fast Selection for PLMB
- 1 = Slow speed used for PLMB (4096 times the timer clock period).
- 0 = Fast speed used for PLMB (256 times the timer clock period). See **PULSE-LENGTH D/A CONVERTERS.**

SM — Slow Mode
- 1 = System runs at 1/16th the normal clock rate (f_{OSC}/32).
- 0 = System runs at normal clock rate (f_{OSC}/2).

WDOG — Watchdog Counter System
- 1 = Watchdog counter system enabled.
- 0 = Watchdog counter system disabled.

NOTE

The reset generated by the watchdog timer is a system reset; thus, the watchdog is disabled after a watchdog reset.

Table 4. External Interrupt Options

INTP	INTN	External Interrupt Options
0	0	Negative Edge and Low-Level Sensitive
0	1	Negative Edge Only
1	0	Positive Edge Only
1	1	Positive and Negative Edge Sensitive

Slow Mode

The slow mode function is controlled by the SM bit in the miscellaneous register ($0C). In slow mode (SM = 1), an extra divide-by-sixteen circuit is added between the oscillator and the internal clock driver. This slows all functions by a factor of 16 (including SCI, A/D, and timer), which is particularly useful in WAIT mode. SM is cleared by external or power-on reset and by STOP mode.

NOTE

If slow mode is enabled while using the A/D, the internal A/D RC oscillator should be turned on.

Watchdog System

The watchdog counter is driven by the 1024 prescaler in the timer and, unless the counter is reset, generates a system reset when it reaches its maximum count (1024 × 8).

The watchdog system is controlled by the WDOG bit in the miscellaneous register ($0C). Writing a one to the bit starts the watchdog or, if it is already started, resets the counter to zero. Writing a zero has no effect; the WDOG bit can only be cleared by external or power-on reset.

The watchdog timer suspends operation during WAIT and resets its count, resuming normal operation following reset.

INTERRUPTS

The MCU can be interrupted four different ways: the three maskable hardware interrupts ($\overline{\text{IRQ}}$, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 9.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 10 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

Four options are available for interrupt triggering sensitivity:

- Negative edge and low level
- Negative edge only
- Positive edge only
- Positive and negative edge

See **Miscellaneous Register ($0C)** for further information.

Figure 11 shows a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

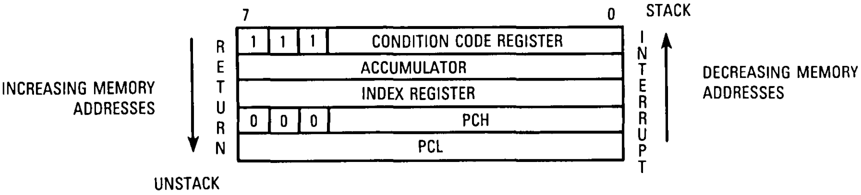
SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and A/D operation (refer to Figure 12).



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 9. Interrupt Stacking Order

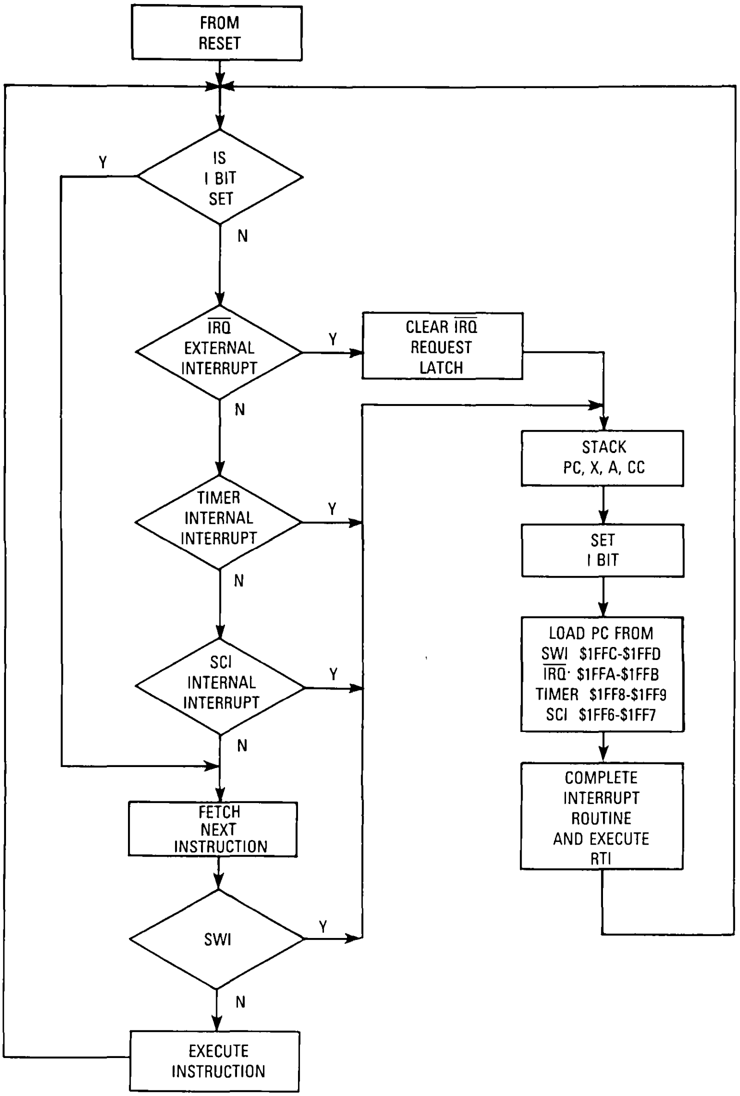
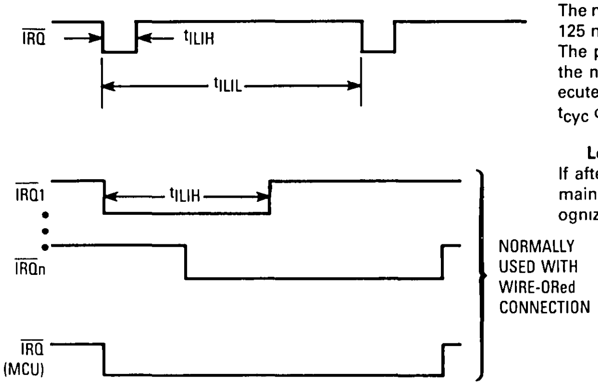

FROM RESET
IS I BIT SET
Y
N
IRQ EXTERNAL INTERRUPT
Y
CLEAR IRQ REQUEST LATCH
N
TIMER INTERNAL INTERRUPT
Y
N
SCI INTERNAL INTERRUPT
Y
N
STACK PC, X, A, CC
SET I BIT
LOAD PC FROM
SWI $1FFC-$1FFD
IRQ· $1FFA-$1FFB
TIMER $1FF8-$1FF9
SCI $1FF6-$1FF7
COMPLETE INTERRUPT ROUTINE AND EXECUTE RTI
FETCH NEXT INSTRUCTION
SWI
Y
N
EXECUTE INSTRUCTION

Figure 10. Reset and Interrupt Processing Flowchart

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI During STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the IRQ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator


$\overline{IRQ}$
t_{ILIH}
t_{ILIL}
$\overline{IRQ1}$
t_{ILIH}
$\overline{IRQn}$
$\overline{IRQ}$ (MCU)
NORMALLY USED WITH WIRE-ORed CONNECTION

Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD}=5 V) or 250 ns (V_{DD}=3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles.

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Figure 11. External Interrupt Mode Diagram

stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

Watchdog During STOP Mode

The STOP instruction is inhibited when the watchdog system is enabled. If a STOP instruction is executed while the watchdog is enabled, a reset occurs that resets the entire MCU.

EEPROM During STOP MODE

The EEPROM is set to read, and the V_{PP1} high-voltage charge pump generator is disabled when stop mode is entered.

PLM During STOP Mode

When the MCU enters stop mode, the PLM outputs remain at their particular level. If power-on or external reset causes the exit from stop mode, the register values are forced to $00.

A/D Converter During STOP Mode

When stop mode is entered with the A/D converter turned on, the A/D clocks are stopped and the A/D converter is disabled for the duration of stop mode, including the t_{PORL}-cycle startup time. If the A/D RC oscillator is used, it will also be disabled.

When leaving STOP mode, after the t_{PORL}-cycle startup time, the A/D converter and A/D RC oscillator resume regular operation. However, a time t_{ADON} is required for the current sources to stabilize. During t_{ADON}, A/D conversion results may be inaccurate.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action and the watchdog system are suspended, but the timer, SCI, PLM, and A/D remain active (refer to Figure 13). An interrupt

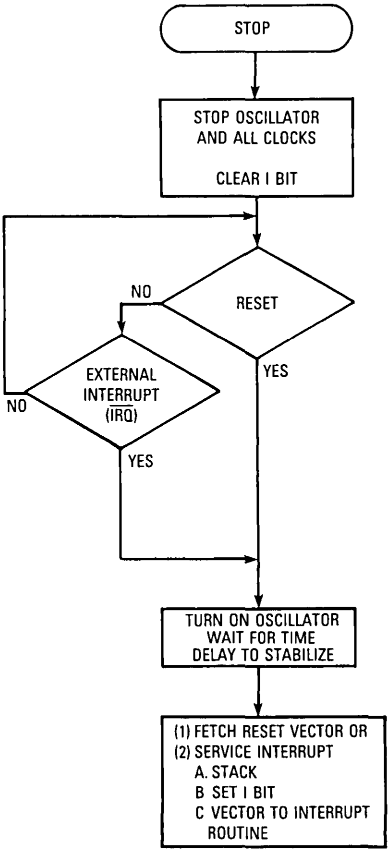

STOP
STOP OSCILLATOR AND ALL CLOCKS
CLEAR I BIT
RESET
NO
YES
EXTERNAL INTERRUPT ($\overline{IRQ}$)
NO
YES
TURN ON OSCILLATOR WAIT FOR TIME DELAY TO STABILIZE
(1) FETCH RESET VECTOR OR
(2) SERVICE INTERRUPT
A. STACK
B SET I BIT
C VECTOR TO INTERRUPT ROUTINE

Figure 12. STOP Function Flowchart

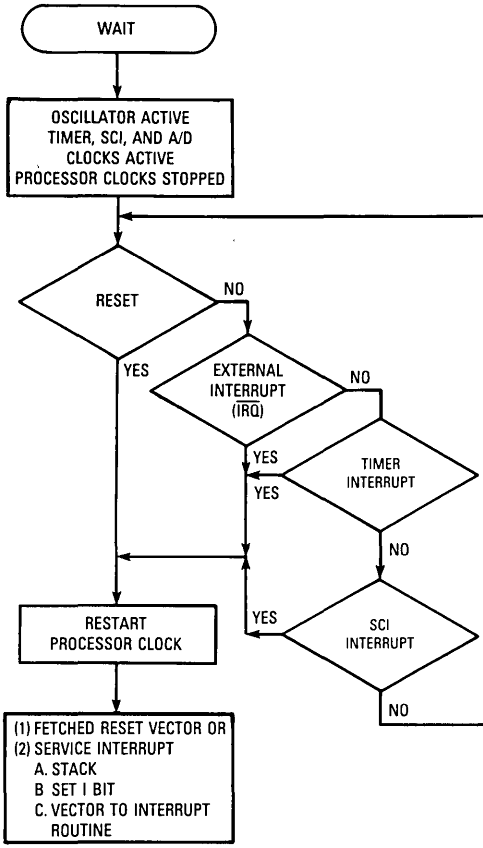

WAIT
OSCILLATOR ACTIVE TIMER, SCI, AND A/D CLOCKS ACTIVE PROCESSOR CLOCKS STOPPED
RESET
NO
YES
EXTERNAL INTERRUPT ($\overline{IRQ}$)
NO
YES
TIMER INTERRUPT
YES
NO
SCI INTERRUPT
YES
NO
RESTART PROCESSOR CLOCK
(1) FETCHED RESET VECTOR OR
(2) SERVICE INTERRUPT
A. STACK
B SET I BIT
C. VECTOR TO INTERRUPT ROUTINE

Figure 13. WAIT Function Flowchart

from the timer, SCI, or an IRQ can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

To achieve proper operation and reduce power consumption, the following points should be set as desired before entering wait mode:

- Timer interrupt enable bits
- A/D control bits
- EEPROM control bits
- SCI enable bits and interrupt enable bits

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements of two input signals while simultaneously generating two output waveforms. Pulse widths can vary from several microseconds to many seconds. The programmable timer works in conjunction with the PLM system to execute two 8-bit D/A PLM conversions, with a choice of two repetition rates. Refer to Figure 14 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

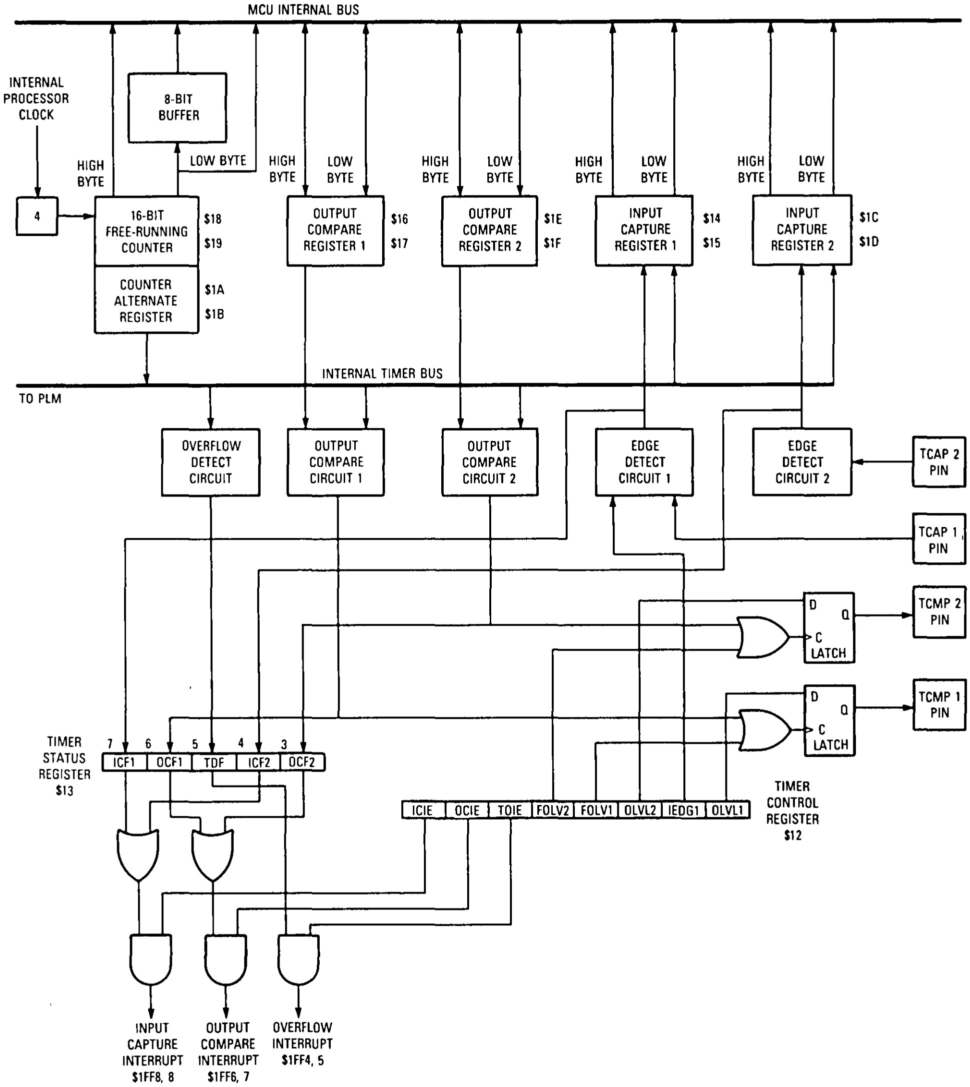

MCU INTERNAL BUS
INTERNAL PROCESSOR CLOCK
8-BIT BUFFER
HIGH BYTE
LOW BYTE
4
16-BIT FREE-RUNNING COUNTER
$18
$19
COUNTER ALTERNATE REGISTER
$1A
$1B
OUTPUT COMPARE REGISTER 1
$16
$17
OUTPUT COMPARE REGISTER 2
$1E
$1F
INPUT CAPTURE REGISTER 1
$14
$15
INPUT CAPTURE REGISTER 2
$1C
$1D
INTERNAL TIMER BUS
TO PLM
OVERFLOW DETECT CIRCUIT
OUTPUT COMPARE CIRCUIT 1
OUTPUT COMPARE CIRCUIT 2
EDGE DETECT CIRCUIT 1
EDGE DETECT CIRCUIT 2
TCAP 2 PIN
TCAP 1 PIN
TCMP 2 PIN
TCMP 1 PIN
D
Q
C
LATCH
TIMER STATUS REGISTER $13
7 6 5 4 3
ICF1 OCF1 TDF ICF2 OCF2
ICIE OCIE TOIE FOLV2 FOLV1 OLVL2 IEDG1 OLVL1
TIMER CONTROL REGISTER $12
INPUT CAPTURE INTERRUPT $1FF8, 8
OUTPUT COMPARE INTERRUPT $1FF6, 7
OVERFLOW INTERRUPT $1FF4, 5

Figure 14. Timer Block Diagram

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, \$18–\$19 (counter register) or \$1A–\$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter (\$19, \$1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) (\$18, \$1A), the LSB (\$19, \$1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register LSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

NOTE

Since the PLM system uses the timer counter, PLM results will be affected when resetting this counter.

OUTPUT COMPARE REGISTERS

There are two output compare registers: output compare register 1 (OCR1) and output compare register 2 (OCR2). The output compare registers can be used for several purposes, such as controlling an output waveform or indicating when a period of time has elapsed All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the four bytes of the output compare registers can be used as storage locations.

NOTE

The same output compare interrupt enable bit is used for the two output compares.

Output Compare Register 1

The output compare register 1 (OCR1) is a 16-bit register, which is made up of two 8-bit registers at locations $16 (most significant byte) and $17 (least significant byte)

The output compare register contents are continually compared with the contents of the free-running counter and, if a match is found, the corresponding output compare flag (OCF1, bit 6 of timer status register $13) is set, and the corresponding output level (OLVL1) bit is clocked to pin TCMP1. The output compare register values and the output level bit should be changed after each successful comparison to control an output waveform or establish a new elapsed timeout. An interrupt can also accompany a successful output compare, provided the corresponding interrupt enable bit, OCIE, is set.

After a processor write cycle to the output compare register 1 containing the most significant byte ($16), the output compare 1 function is inhibited until the least significant byte ($17) is also written. The user must write both bytes (locations) if the most significant byte is written first. A write made only to the least significant byte ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register 1 is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register 1 without affecting the other byte. The output level (OLVL1) bit is clocked to the corresponding output level register and then to the TCMP1 pin, regardless of whether the output compare flag (OCF1) is set or clear

Output Compare Register 2

The output compare register 2 (OCR2) is a 16-bit register, which is made up of two 8-bit registers at locations $1E (most significant byte) and $1F (least significant byte). The function of OCR2 is identical to OCR1, requiring only changes of the register locations and control bits in the timer status register ($13) to make the OCR1 description apply to OCR2.

SOFTWARE FORCE COMPARE

The MCU provides a force compare capability to facilitate fixed frequency generation as well as other applications. Bit 3 (FOLV1 for OCR1) and bit 4 (FOLV2 for OCR2) in the timer control register ($12) implement this force compare. Writing a one to these bits causes the OLVL1 or OLVL2 values to be copied to the respective output registers (TCMP1 or TCMP2 pins). Internal logic allows a single instruction to change OLVL1 and OLVL2 and cause a forced compare with the new values of OLVL1 and OLVL2.

NOTE

A software force compare, which affects the corresponding output pin TCMP1 or TCMP2, does not affect the compare flag; thus, it does not generate an interrupt.

INPUT CAPTURE REGISTERS

There are two input capture registers: input capture register 1 (ICR1) and input capture register 2 (ICR2).

NOTE

The same input capture interrupt enable bit (ICIE) is used for the two input capture registers.

Input Capture Register 1

Two 8-bit registers that make up the 16-bit input capture register 1 (ICR1) are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition that triggers the counter transfer is defined by the corresponding input edge bit (IEDG1) Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal-bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition, regardless of whether the input capture flag (ICF1) is set or clear. The input capture register always contains the free-running counter value, which corresponds to the most recent input capture.

After a read of the input capture register 1 ($14) most significant byte, the counter transfer is inhibited until the least significant byte ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register 1 least significant byte ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

Input Capture Register 2

The input capture register 2 (ICR2) is a 16-bit register that is composed of two 8-bit registers at locations $1C (most significant byte) and $1D (least significant byte). Input capture register 2 functions identically to input capture register 1, except that only negative edge sensitivity is available. By substituting the appropriate bits in the timer status register ($13) and substituting register locations, the ICR1 description applies to ICR2.

TIMER CONTROL REGISTER (TCR) $12

The TCR is an 8-bit read/write register, illustrated below with a definition of each bit.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
RESET	0	0	0	0	0	0	U	0

(U Unaffected)

ICIE — Input Capture Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

FOLV2 — Force Output Compare 2
- 1 = Forces the OLVL2 bit to the corresponding output latch
- 0 = No effect

FOLV1 — Force Output Compare 1
- 1 = Forces the OLVL1 bit to the corresponding output latch
- 0 = No effect

OLVL2 — Output Level 2
- 1 = The value of the output level 2 bit, which is copied to the output level latch by the next successful output compare 2, appears at TCMP2
- 0 = No effect

IEDG1 — Input Edge 1

Value of input edge determines which level transition on TCAP1 pin will trigger free-running counter transfer to the input capture register.
- 1 = Positive edge
- 0 = Negative edge

OLVL1 — Output Level 1

Value of output level 1 bit, which is copied into output level register by the next successful output compare 1, will appear on the TCMP1 pin.
- 1 = High output
- 0 = Low output

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits. Bits 0–4 always read zero.

	7	6	5	4	3	2	1	0
	ICF1	OCF1	TOF	ICF2	OCF2	—	—	—
RESET	U	U	U	U	U	—	—	—

ICF1 — Input Capture Flag 1
- 1 = Flag set when selected polarity edge is sensed by input capture edge detector
- 0 = Flag cleared when TSR and input capture 1 low register ($15) are accessed

OCF1 — Output Capture Flag 1
- 1 = Flag set when output compare register contents match the free-running counter contents
- 0 = Flag cleared when TSR and output compare 1 low register ($17) are accessed

TOF — Timer Overflow Flag
- 1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
- 0 = Flag cleared when TSR and counter low register ($19) are accessed

ICF2 — Input Capture Flag 2
- 1 = Flag set when selected polarity edge is sensed by input capture 2 edge detector
- 0 = Flag cleared when TSR and input capture 2 low register ($1D) are accessed

OCF2 — Output Capture Flag 2
- 1 = Flag set when output compare register contents match the free-running counter contents
- 0 = Flag cleared when TSR and output compare low register 2 ($1F) are accessed

Bits 0–2 — Not Used

Can read either zero or one.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If reset is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags or wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If reset is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit. A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if 1) the timer status register is read or written when TOF is set, and 2) the least significant byte of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate prescaler. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Different baud rates possible for transmit and receive
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 15.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

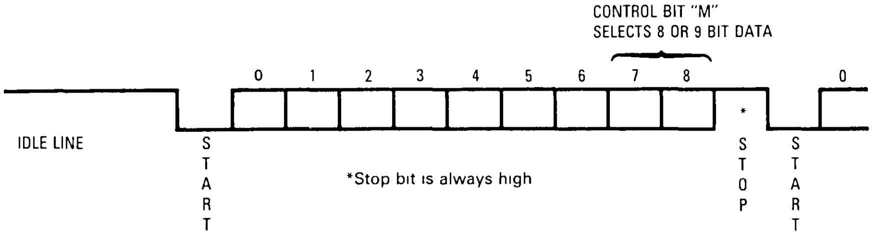


Figure 15. Data Format

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 16); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

SCI SYNCHRONOUS TRANSMISSION

The SCI transmitter allows a one-way synchronous transmission, with the SCLK pin as the clock output. No clock is sent to the SCLK pin during the stop and start bits. The LCL bit (SSCR1) controls whether clocks are active during the last valid data bit (address mark). The CPOL bit selects clock polarity, and the CPHA bit selects the phase of the external clock. During idle, preamble, and send break, the external SCLK clock is not active.

These options allow the SCI to control serial peripherals consisting of shift registers without losing any function of the SCI transmitter These options do not affect the SCI receiver, which is independent of the transmitter.

The SCLK pin works in conjunction with the TDO pin. When the SCI transmitter is disabled, the SCLK and the TDO pins assume a high-impedance state.

NOTE

The LBCL, CPOL, and CPHA bits must be selected before the transmitter is enabled to ensure that the clocks function correctly. These bits should not be changed while the transmitter is enabled.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock (if the same baud rate is used for transmit and receive).

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 16. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates for the transmitter and receiver.

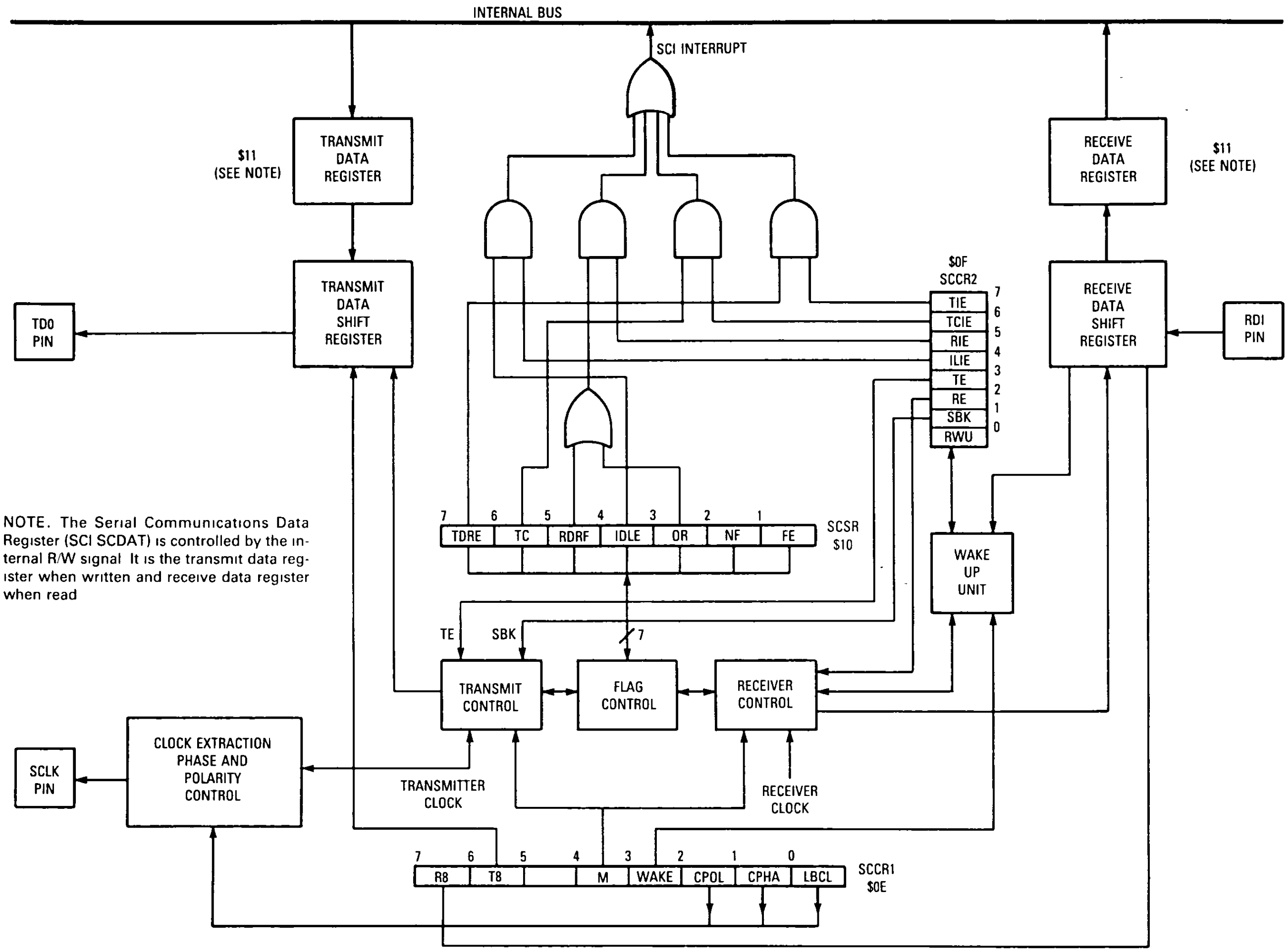
Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.


INTERNAL BUS
SCI INTERRUPT
$11
(SEE NOTE)
TRANSMIT DATA REGISTER
RECEIVE DATA REGISTER
$11
(SEE NOTE)
TDO PIN
TRANSMIT DATA SHIFT REGISTER
$0F
SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
7
6
5
4
3
2
1
0
RECEIVE DATA SHIFT REGISTER
RDI PIN
7
6
5
4
3
2
1
TDRE
TC
RDRF
IDLE
OR
NF
FE
SCSR
$10
WAKE UP UNIT
TE
SBK
7
TRANSMIT CONTROL
FLAG CONTROL
RECEIVER CONTROL
SCLK PIN
CLOCK EXTRACTION PHASE AND POLARITY CONTROL
TRANSMITTER CLOCK
RECEIVER CLOCK
7
6
5
4
3
2
1
0
R8
T8
M
WAKE
CPOL
CPHA
LBCL
SCCR1
$0E

NOTE. The Serial Communications Data Register (SCI SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 16. SCI Block Diagram

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET·	U	U	U	U	U	U	U	U

(U = Unaffected)

As shown in Figure 16, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $0E

The SCCR1 provides control bits that determine word length, select the wake-up method, and control the options to output the transmitter clocks for synchronous transmissions.

	7	6	5	4	3	2	1	0
	R8	T8	—	M	WAKE	CPOL	CPHA	LBCL
RESET.	U	U	—	U	U	U	U	U

(U = Unaffected)

R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit

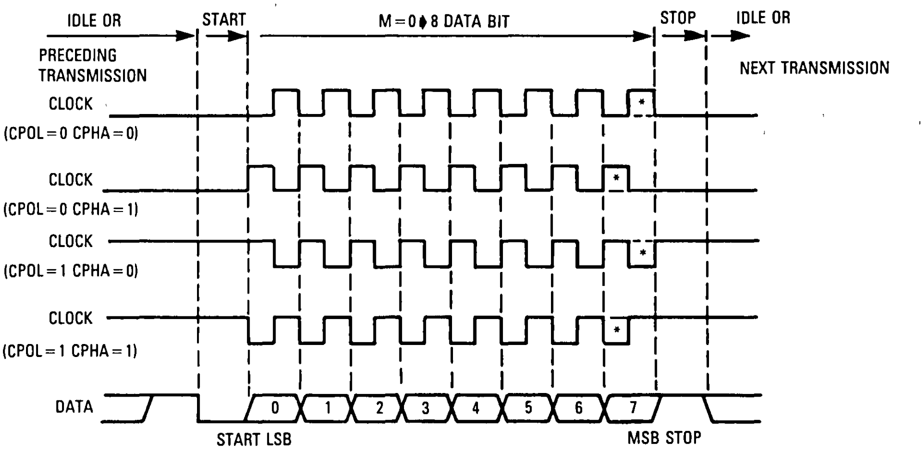
WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most significant bit)
0 = Idle line condition

CPOL — Clock Polarity
Selects the clock polarity sent to the SCLK pin.
1 = Steady state high outside the transmission window
0 = Steady state low outside the transmission window
The CPOL bit should not be changed with the transmitter active.

CPHA — Clock Phase
Selects the clock phase sent to the SCLK pin.
1 = SCLK line activated at the beginning of the data bit
0 = SCLK line activated in the middle of the data bit (see Figures 17 and 18)
The CPHA bit should not be changed with the transmitter active.

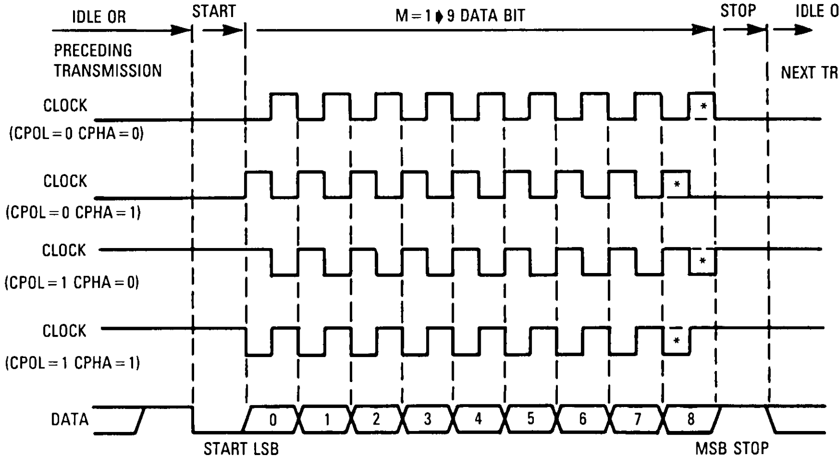
LBCL — Last Bit Clock
Selects whether the clock associated with the last data bit transmitted is output to the SCLK pin.
1 = Last data bit output
0 = Last data bit not output
The last data bit is the eighth or ninth bit, depending on whether an 8- or 9-bit format is used (see Table 5). The LBCL bit should not be changed while the transmitter is enabled.

Bit 5 — Not used
Can be 1 or 0.



*LBCL Bit Controls Last Data Clock

Figure 17. SCI Data Clock Timing Diagram (M = 0)



*LBCL Bit Controls Last Data Clock

Figure 18. SCI Data CLock Timing Diagram (M=1)

Table 5. SCI Clock on SCLK Pin

Data Format	M Bit	LBCL Bit	Number of Clocks on SCLK Pin
8 Bit	0	0	7
8 Bit	0	1	8
9 Bit	1	0	8
9 Bit	1	1	9

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $0F

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt enabled, provided TDRE is set
0 = TDRE interrupt disabled

TCIE — Transmit Complete Interrupt Enable
1 = SCI interrupt enabled, provided TC is set
0 = TC interrupt disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt enabled, provided OR or RDRF is set
0 = RDRF and OR interrupts disabled

ILIE — Idle Line Interrupt Enable
1 = SCI interrupt enabled, provided IDLE is set
0 = Idle interrupt disabled

TE — Transmit Enable
1 = Transmit shift register output is applied to the TD0 line, and the corresponding clocks are applied to the SCLK pin. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TD0 line becomes a high-impedance line.

RE — Receive Enable
1 = Receiver shift register input is applied to the RDI line.
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.

RWU — Receiver Wake-Up
- 1 = Places receiver in sleep mode and enables wake-up function
- 0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
  Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)

SBK — Send Break
- 1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
- 0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET	1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
- 1 = TDR contents transferred to the transmit data shift register
- 0 = TDR still contains data. TDRE is cleared by reading the SCSR followed by a write to the TDR.

TC — Transmit Complete
- 1 = Indicates end of data frame, preamble, or break condition has occurred if:
  1. TE = 1, TDRE = 1, and no pending data, preamble or break is to be transmitted; or
  2. TE = 0 and the data preamble or break (in the transmit shift register) has been transmitted.
- 0 = TC bit cleared by reading the SCSR followed by a write to the TDR

  The TC bit is a status register that indicates one of the above conditions has occurred. It does not inhibit the transmitter in any way.

RDRF — Receive Data Register (RDR) Full
- 1 = Receive data shift register contents transferred to the RDR
- 0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR followed by a read of the RDR

IDLE — Idle Line Detect
- 1 = Indicates receiver has detected an idle line
- 0 = IDLE is cleared by reading the SCSR followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error
- 1 = Indicates receive data shift register data is ready to be sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
- 0 = OR is cleared by reading the SCSR followed by a read of the RDR.

NF — Noise Flag
- 1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
- 0 = NF is cleared by reading the SCSR followed by a read of the RDR.

FE — Framing Error
- 1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.
- 0 = FE is cleared by reading the SCSR followed by a read of the RDR.

Bit 0 — Not used
Can read either one or zero

Baud Rate Register $0D

The baud rate register selects the SCI transmitter and receiver baud rate. The SCP1 and SCP0 prescaler bits are used in conjunction with the SCR2–SCR0 bits to generate the receiver baud rate and in conjunction with the SCT2–SCT0 baud rate bits to generate the transmitter baud rate.

Tables 6 and 7 tabulate the divide chain used to obtain the baud-rate clock (transmit or receive clock). The actual divider chain is controlled by the combined SCP1–SCP0 and SCR2–SCR0 or SCT2–SCT0 bits in the baud rate register. The dividend frequencies shown in Table 6 represent the final baud rate that results from prescaler division only (SCR or SCT bits all zero). Table 7 lists the prescaler output frequency divided by the action of the SCR or SCT bits.

For example, assume that a 9600-Hz baud rate is desired from a 2.4576-MHz system clock crystal. The prescaler bits could be set for either a divide-by-one or divide-by-four. If a divide-by-four prescaler is used, then the SCR and SCT bits must be set for divide-by-two. The same result, using the same crystal frequency, can be obtained with a prescaler divide-by-one and SCR and SCT bit divide-by-eight.

	7	6	5	4	3	2	1	0
	SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0
RESET	0	0	U	U	U	U	U	U

SCP1–SCP0 — SCI Prescaler Bit 1 and 0

These two prescaler bits are used to increase the range of standard baud rates controlled by the SCT2–SCT0 and SCR2–SCR0 bits. Prescaler internal processor clock division versus bit levels are listed in Table 6.

SCR2–SCT0 — SCI Transmit Baud Rate Selection Bits

These three bits, taken in conjunction with bits SCP1–SCP0, are used to select the SCI transmit baud rate. Baud rates versus bit levels are listed in Table 7.

Table 6. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	0		4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43 691 kHz	41.666 kHz	25.60 kHz	20 833 kHz	19 20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15 625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 6 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs

Table 7. Transmit Baud Rate Output for a Given Prescaler Output

SCR/T Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131 072 kHz	32 768 kHz	76 80 kHz	19 20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38 40 kHz	9600 Hz	4800 Hz
0	1	0	4	32 768 kHz	8 192 kHz	19 20 kHz	4800 Hz	2400 Hz
0	1	1	8	16 384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8 192 kHz	2 048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2 048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1 024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 7 illustrates how the SCI select bits can be used to provide lower transmitter or receiver baud rates by further dividing the prescaler output frequency The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

SCR2–SCR0 — SCI Receive Baud Rate Selection Bits

These three bits, taken in conjunction with bits SCP1–SCP0, are used to select the SCI receive baud rate. Baud rates versus bit levels are listed in Table 7.

PULSE-LENGTH D/A CONVERTERS

The pulse-length D/A converter (PLM) works in conjunction with the timer to execute two 8-bit conversions with a choice of two repetition rates. The outputs are pulse-length modulated signals whose duty-cycle ratio may be modified. These signals can be used directly as PLMS, or the filtered average values can be used as general-purpose analog outputs.

Registers PLMA and PLMB contain the pulse-length values for the two PLMs. A value of $00 results in a continuously low output from the D/A. A value of $80 results in a 50-percent duty-cycle output, and a value of $FF gives an output that is a logic 1 for 255/256 of the cycle. When the MCU writes to the PLMA or PLMB register, the D/A picks up the new value at the end of a complete conversion cycle so that a monotonic change in the dc component of the output results. This monotonic change avoids overshoots or vicious starts (a vicious start is an output that gives totally erroneous output during the first cycle following an update of the registers). WAIT mode does not affect the output waveform of the D/A converters.

NOTE

Since the PLM system uses the timer counter, PLM results will be affected while resetting the timer counter.

Figure 19 shows a block diagram of the PLM system

PLMA ($0A)

7	6	5	4	3	2	1	0
PLMA7	PLMA6	PLMA5	PLMA4	PLMA3	PLMA2	PLMA1	PLMA0

RESET.

0	0	0	0	0	0	0	0

PLMB ($0B)

7	6	5	4	3	2	1	0
PLMB7	PLMB6	PLMB5	PLMB4	PLMB3	PLMB2	PLMB1	PLMB0

RESET·

0	0	0	0	0	0	0	0

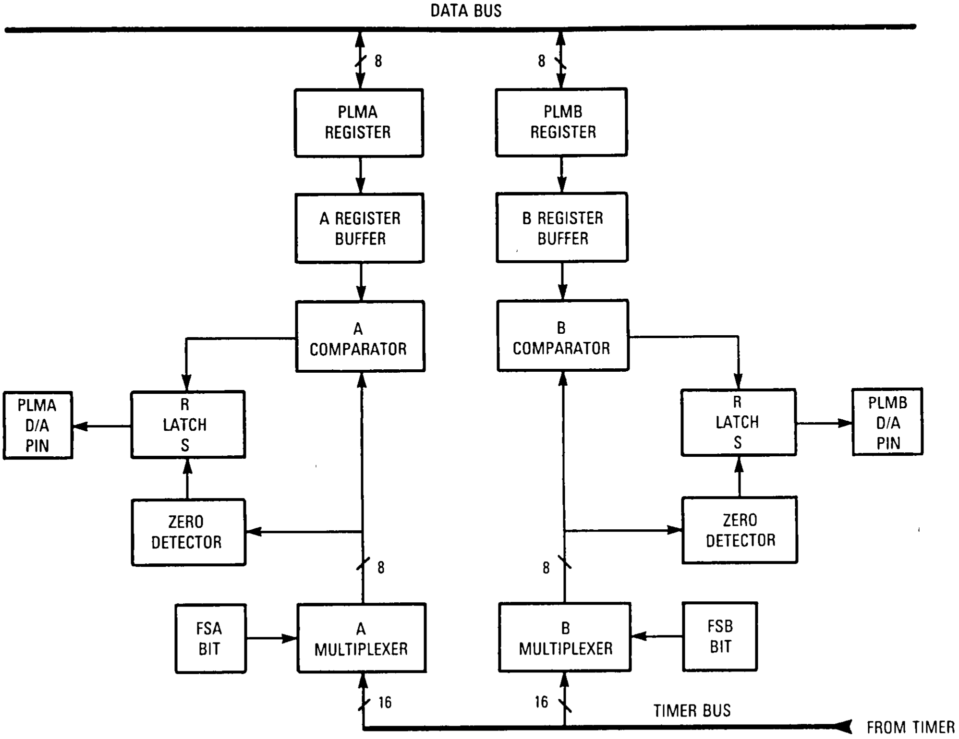

DATA BUS
8
8
PLMA REGISTER
PLMB REGISTER
A REGISTER BUFFER
B REGISTER BUFFER
A COMPARATOR
B COMPARATOR
PLMA D/A PIN
R LATCH S
R LATCH S
PLMB D/A PIN
ZERO DETECTOR
ZERO DETECTOR
8
8
FSA BIT
A MULTIPLEXER
B MULTIPLEXER
FSB BIT
16
16
TIMER BUS
FROM TIMER

Figure 19. PLM Block Diagram

Miscellaneous ($0C)

7	6	5	4	3	2	1	0
—	—	—	—	SFA	SFB	—	—
RESET							
—	—	—	—	0	0	—	—

SFA — Slow/Fast Control for PLMA Clock
- 1 = Slow speed of PLMA used (4096 times the timer clock period)
- 0 = Fast speed of PLMA used (256 times the timer clock period)

SFB — Slow/Fast Control for PLMB Clock
- 1 = Slow speed of PLMB used (4096 times the timer clock period)
- 0 = Fast speed of PLMB used (256 times the timer clock period)

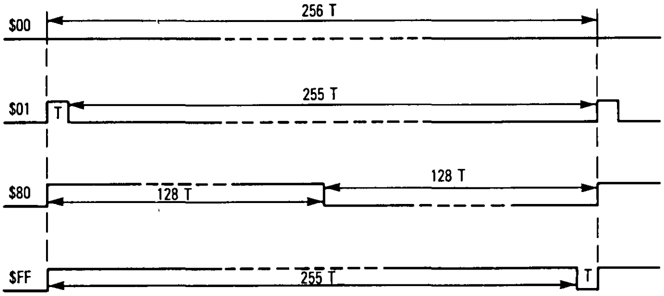
NOTE

The highest speed of the PLM system corresponds to the frequency of the TOF bit being set, multiplied by 256. The slowest speed of the PLM system corresponds to the frequency of the TOF bit being set, multipled by 16.

The SFA and SFB bits are not double buffered; therefore, these bits must be selected before writing to either PLM register to avoid temporary wrong values from the PLM outputs. Figure 20 shows some examples of the PLM output waveforms.

A/D CONVERTER

The A/D converter system consists of an 8-bit successive approximation converter and a 16-channel multiplexer. Eight of the channels are available for output, and the other eight channels are dedicated to internal test functions. There is one 8-bit result data register (address $08) and one 8-bit status/control register (address $09). The reference supply for the converter uses dedicated input pins instead of the power supply lines, because drops caused by loading in the power supply lines would degrade the accuracy of the A/D conversion. An internal RC oscillator is available if the bus speed is low enough to degrade the A/D accuracy. An ADON bit allows the A/D to be switched off to reduce power consumption, which is particularly useful in the WAIT mode.



T = 4 CPU Clocks Fast Mode
T = 64 CPU Clocks Slow Mode

Figure 20. PLM Output Waveform Examples

For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}. An input voltage greater than or equal to V_{RH} converts as $FF (full scale) with no overflow indication. An input voltage equal to V_{RL} converts as $00. The conversion is monotonic with no missing codes.

A/D STATUS/CONTROL REGISTER ($09)

7	6	5	4	3	2	1	0
COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0

RESET

0	0	0	0	0	0	0	0

COCO — Conversion Complete
- 1 = Conversion is complete; a new result can be read from the result data register ($08).
- 0 = No conversion since last reset

ADRC — A/D RC Oscillator Control
- 1 = A/D uses RC clock
- 0 = A/D uses CPU clock

When the RC oscillator is turned on, it requires a time t_{ADRC} to stabilize, and results can be inaccurate during this time.

ADON — A/D On
- 1 = A/D enabled
- 0 = A/D disabled

When the A/D is turned on, it requires a time t_{ADON} for the current sources to stabilize, and results can be inaccurate during this time.

CH3–CH0 — Channel 3 through Channel 0

These bits select the A/D channel assignment (see Table 8).

NOTE

Using one or more pins of PD0/AN0–PD7/AN7 as analog inputs does not affect the ability to use port D inputs as digital inputs. However, using port D for digital inputs during an analog conversion sequence may inject noise on the analog inputs and reduce the accuracy of the A/D result.

Table 8. A/D Channel Assignments

CH3	CH2	CH1	CH0	Channel Selected
0	0	0	0	AN0, Port D Bit 0
0	0	0	1	AN1, Port D Bit 1
0	0	1	0	AN2, Port D Bit 2
0	0	1	1	AN3, Port D Bit 3
0	1	0	0	AN4, Port D Bit 4
0	1	0	1	AN5, Port D Bit 5
0	1	1	0	AN6, Port D Bit 6
0	1	1	1	AN7, Port D Bit 7
1	0	0	0	V_{RH} Pin (High)
1	0	0	1	$((V_{RH})+(V_{RL}))/2$
1	0	1	0	V_{RL} Pin (Low)
1	0	1	1	V_{RL} Pin (Low)
1	1	0	0	V_{RL} Pin (Low)
1	1	0	1	V_{RL} Pin (Low)
1	1	1	0	V_{RL} Pin (Low)
1	1	1	1	V_{RL} Pin (Low)

Performing a digital read of port D with levels other than V_{DD} or V_{SS} on the inputs causes greater than normal power dissipation during the read and may give erroneous results.

INSTRUCTION SET

The MCU instructions can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X A ← X · A		
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register		
Condition Codes	H Cleared I Not affected N Not affected Z Not affected C Cleared		
Source Form(s)	MUL		
Addressing Mode	Cycles	Bytes	Opcode
Inherent	11	1	$42

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 9 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510

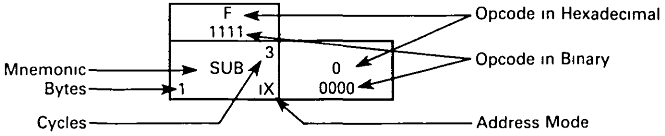
Table 9. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
HI / LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LOW
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.5 to V_{DD} +0 5	V
Bootstrap Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} − 0 5 to 2 × V_{DD} + 0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC805B6FN (Standard) MC68HC805B6CFN (Extended) MC68HC805B6MFN (Automotive)	T_A	T_L to T_H 0 to +70 −40 to +85 −40 to +125	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e g , either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Plastic Leaded Chip Carrier (PLCC)	θ_{JA}	 40 50	°C/W

$V_{DD} = 4.5$ V

Pins	R1	R2	C
PA7–PA0, PB7–PB0, PC7–PC0, TCMP1 TCMP2	3.26 kΩ	2.38 kΩ	50 pF
TDO, SCLK, PLMA, PLMB	1.9 kΩ	2.26 kΩ	200 pF

$V_{DD} = 3.0$ V

Pins	R1	R2	C
PA7–PA0, PB7–PB0, PC7–PC0, TCMP1, TCMP2	10.91 kΩ	6.32 kΩ	50 pF
TDO, SCLK, PLMA, PLMB	6 kΩ	6 kΩ	200 pF

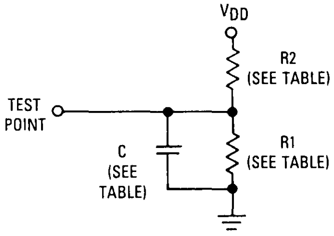


Figure 21. Equivalent Test Load

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ μA	V_{OL} V_{OH}	— $V_{DD}-0.1$	— —	0.1 —	V
Output High Voltage	V_{OH}				V
(I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP1, TCMP2		$V_{DD}-0.8$	$V_{DD}-0.4$	—	
(I_{Load} = 1.6 mA) TDO, SCLK, PLMA, PLMB		$V_{DD}-0.8$	$V_{DD}-0.4$	—	
Output Low Voltage	V_{OL}				V
(I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP1, TCMP2, PLMA, PLMB, TDO, SCLK		—	0.1	0.4	
$\overline{\text{RESET}}$		—	0.4	1.0	
Input High Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7–PD0, TCAP1, TCAP2, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, RDI	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7–PD0, TCAP1, TCAP2, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, RDI	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
EEPROM6 Programming Voltage	V_{PP6}	V_{DD}	—	19	V
Supply Current (see Notes)	I_{DD}				
RUN (SM = 0)		—	3.5	9	mA
RUN (SM = 1, t_{cyc} = 8 μs)		—	0.5	2	mA
WAIT (SM = 0)		—	1	4	mA
WAIT (SM = 1, t_{cyc} = 8 μs)		—	0.35	1	mA
STOP					
0 to 70°C (Standard)		—	2	10	μA
−40 to 85°C (Extended)		—	—	20	μA
−40 to 125°C (Automotive)		—	—	50	μA
I/O Ports Hi-Z Leakage Current PA7-PA0, PB7-PB0, PC7-PC0, TDO, $\overline{\text{RESET}}$, SCLK	I_{IL}	—	±0.2	±1	μA
Input Current	I_{in}				μA
$\overline{\text{IRQ}}$, TCAP1, TCAP2, OSC1, RDI		—	±0.2	±1	
PD0/AN0–PD7/AN7 (A/D off)		—	±0.2	±1	
PD0/AN0–PD7/AN7 (A/D on)		—	±10	TBD	
Capacitance					pF
Ports (as Input or Output), $\overline{\text{RESET}}$	C_{out}	—	—	12	
TDO, SCLK	C_{out}	—	—	12	
$\overline{\text{IRQ}}$, TCAP1, TCAP2, OSC1, RDI	C_{in}	—	—	8	
PD0/AN0–PD7/AN7 (A/D off)	C_{in}	—	12	TBD	
PD0/AN0–PD7/AN0 (A/D on)	C_{in}	—	22	TBD	

TBD = To be determined.

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (TE = RE = 0) If SCI active (TE = RE = 1) add 10% current draw
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.0 MHz), all inputs 0.2 V from rail, no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V
6. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \le 10.0$ µA	V_{OL}	—	—	0.1	V
	V_{OH}	$V_{DD}-0.1$	—	—	
Output High Voltage	V_{OH}				V
(I_{Load} = 0.2 mA) PA7–PA0, PB7–PB0, PC7–PC0, TCMP1, TCMP2		$V_{DD}-0.3$	$V_{DD}-0.1$	—	
(I_{Load} = 0.4 mA) TDO, SCLK, PLMA, PLMB		$V_{DD}-0.3$	$V_{DD}-0.1$	—	
Output Low Voltage	V_{OL}				V
(I_{Load} = 0.4 mA) PA7–PA0, PB7–PB0, PC7–PC0, TCMP1, TCMP2, PLMA, PLMB, TDO, SCLK		—	0.1	0.3	
$\overline{\text{RESET}}$		—	0.2	0.6	
Input High Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7–PD0, TCAP1, TCAP2, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, RDI	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7–PD0, TCAP1, TCAP2, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, RDI	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
EEPROM6 Programming Voltage	V_{PP6}	V_{DD}	—	19	V
Supply Current (see Notes)	I_{DD}				
RUN (SM = 0)		—	1.2	5	mA
RUN (SM = 1, t_{cyc} = 8 µs)		—	0.2	1	mA
WAIT (SM = 0)		—	0.4	2	mA
WAIT (SM = 1, t_{cyc} = 8 µs)		—	0.15	0.5	mA
STOP					µA
0 to 70°C (Standard)		—	1	10	µA
−40 to +85°C (Extended		—	—	10	µA
−40 to 125°C (Automotive)		—	—	30	
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB0, PC7–PC0, TDO, $\overline{\text{RESET}}$, SCLK	I_{IL}	—	±0.2	±10	µA
Input Current	I_{in}				µA
$\overline{\text{IRQ}}$, TCAP1, TCAP2, OSC1, RDI		—	±0.2	±1	
PD0/AN0–PD7/AN7 (A/D off)		—	±0.2	±1	
PD0/AN0–PD7/AN7 (A/D on)		—	±10	TBD	
Capacitance					pF
Ports (as Input or Output), $\overline{\text{RESET}}$, TDO	C_{out}	—	—	12	
TDO, SCLK	C_{out}	—	—	12	
$\overline{\text{IRQ}}$, TCAP1, TCAP2, OSC1, RDI	C_{in}	—	—	8	
PD0/AN0–PD7/AN7 (A/D off)	C_{in}	—	12	TBD	
PD0/AN0–PD7/AN7 (A/D on)	C_{in}	—	22	TBD	

TBD = To be determined.

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (TE = RE = 0). If SCI active (TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}. All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = $V_{DD}-0.2$ V.
6. Wait I_{DD} is affected linearly by the OSC2 capacitance.

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc)

Characteristic	Parameter	Min	Max	Unit
Resolution	Number of bits resolved by the A/D	8	—	Bit
Non-Linearity	Maximum deviation from the best straight line through the A/D transfer characteristics (V_{RH} = V_{DD} and V_{RL} = 0 V)	—	± ½	LSB
Quantization Error	Uncertainty due to converter resolution	—	± ½	LSB
Absolute Accuracy	Difference between the actual input voltage and the full-scale equivalent of the binary code output code for all errors	—	± 1	LSB
Conversion Range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} − 0.1	V_{RH}	V
Conversion Time	Total time to perform a single analog-to-digital conversion a. External Clock (XTAL, EXTAL) b. Internal RC oscillator	 — —	 32 32	 t_{cyc} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero-Input Reading	Conversion result when V_{in} = V_{RL}	00	—	Hex
Full-Scale Reading	Conversion result when V_{in} = V_{RH}	—	FF	Hex
Sample Acquisition Time (see Note 1)	Analog input acquisition sampling a. External Clock (XTAL, EXTAL) b. Internal RC oscillator	 — —	 12 12	 t_{cyc} µs
Sample/Hold Capacitance	Input capacitance on PD0/AN0–PD7/AN7	—	12	pF
Input Leakage (see Note 2)	Input leakage on A/D pins PD0/AN0–PD7/AN7, V_{RL}, V_{RH}	— —	1 1	µA

NOTES:

1. Source impedances greater than 10K ohm will adversely affect internal RC charging time during input sampling.
2. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	4.2	
External Clock Option		dc	4.2	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc}/2$)		—	2.1	
External Clock ($f_{osc}/2$)		dc	2.1	
Cycle Time (see Figure 23)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 23)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
External RESET Input Pulse Width (see Figure 23)	t_{RL}	1.5	—	t_{cyc}
Power-On RESET Output Pulse Width	t_{PORL}			t_{cyc}
4064 Cycle Option		4064	—	
16 Cycle Option		16	—	
Watchdog RESET Output Pulse Width	t_{DOGL}	1.5	—	t_{cyc}
Watchdog Time-Out	t_{DOG}	6144	7168	t_{cyc}
EEPROM1 Byte Erase Time	t_{ERA}			ms
0 to 70°C (Standard)		10	—	
0 to 85°C (Extended)		10	—	
0 to 125°C (Automotive)		10	—	
EEPROM1 Byte Programming Time	t_{PROG}			ms
0 to 70°C (Standard)		10	—	
0 to 85°C (Extended)		10	—	
0 to 125°C (Automotive)		20	—	
EEPROM6 Bulk Erase Time	t_{ERA6}			ms
0 to 70°C (Standard)		10	—	
0 to 85°C (Extended)		10	—	
0 to 125°C (Automotive)		10	—	
EEPROM6 Byte Programming Time	T_{PROG6}			ms
0 to 70°C (Standard)		10	—	
0 to 85°C (Extended)		10	—	
0 to 125°C (Automotive)		20	—	
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 22	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (see Figure 22)	t_{TL}, t_{TL}	***	—	t_{cyc}
Interrupt Pulse Width (Edge-Triggered) (see Figure 11)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 11)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

NOTES:

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}

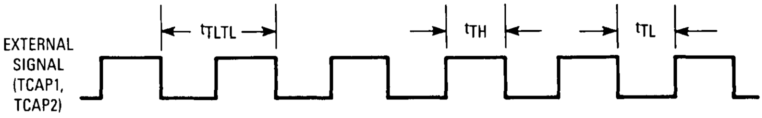


Figure 22. Timer Relationship

CONTROL TIMING (V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

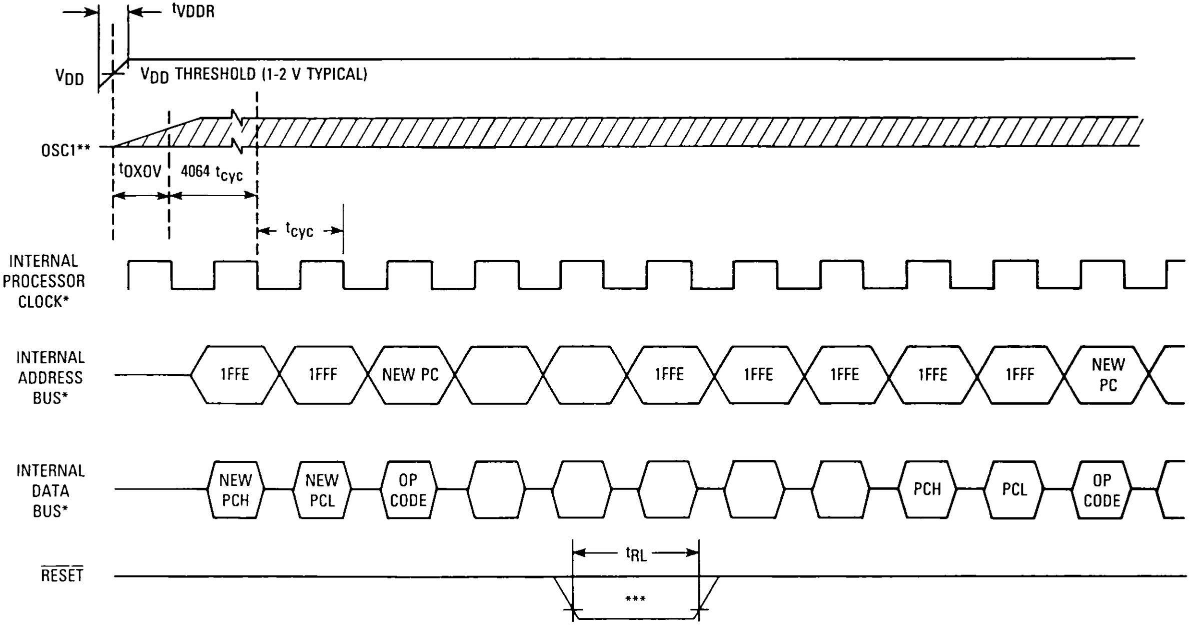
Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal (f_{osc}/2)		—	1.0	
External Clock (f_{osc}/2)		dc	1.0	
Cycle Time (see Figure 23)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 23)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
External RESET Input Pulse Width (see Figure 23)	t_{RL}	1.5	—	t_{cyc}
Power-On RESET Output Pulse Width	t_{PORL}			t_{cyc}
4064 Cycle Option		4064	—	
16 Cycle Option		16	—	
Watchdog RESET Output Pulse Width	t_{DOGL}	1.5	—	t_{cyc}
Watchdog Time-Out	t_{DOG}	6144	7168	t_{cyc}
EEPROM1 Byte Erase Time	t_{ERA}			ms
0 to 70°C (Standard)		30	—	
0 to 85°C (Extended)		TBD	—	
0 to 125°C (Automotive)		TBD	—	
EEPROM1 Byte Programming Time	t_{PROG}			ms
0 to 70°C (Standard)		30	—	
0 to 85°C (Extended)		TBD	—	
0 to 125°C (Automotive)		TBD	—	
EEPROM6 Bulk Erase Time	t_{ERA6}			ms
0 to 70°C (Standard)		30	—	
0 to 85°C (Extended)		TBD	—	
0 to 125°C (Automotive)		TBD	—	
EEPROM6 Byte Programming Time	t_{PROG6}			ms
0 to 70°C (Standard)		30	—	
0 to 85°C (Extended)		TBD	—	
0 to 125°C (Automotive)		TBD	—	
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 22)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 22)	t_{TL}, t_{TL}	***	—	t_{cyc}
Interrupt Pulse Width (Edge-Triggered) (see Figure 11)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 11)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

NOTES

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.


t_{VDDR}
V_{DD}
V_{DD} THRESHOLD (1-2 V TYPICAL)
OSC1**
t_{OXOV}
4064 t_{cyc}
t_{cyc}
INTERNAL PROCESSOR CLOCK*
INTERNAL ADDRESS BUS*
1FFE
1FFF
NEW PC
1FFE
1FFE
1FFE
1FFE
1FFF
NEW PC
INTERNAL DATA BUS*
NEW PCH
NEW PCL
OP CODE
PCH
PCL
OP CODE
$\overline{\text{RESET}}$
t_{RL}
***

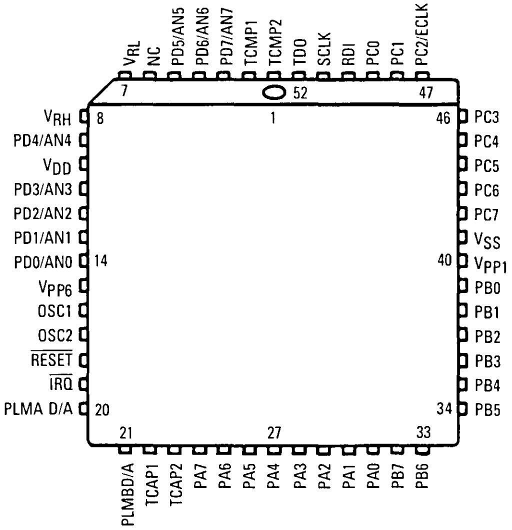
*Internal timing signal and bus information not available externally
**OSC1 line is not meant to represent frequency It is only used to represent time
***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 23. Power-On Reset and $\overline{\text{RESET}}$

ORDERING INFORMATION

Package Type	Temperature	Order Number
PLCC (FN Suffix)	0°C to +70°C −40°C to +85°C −40°C to +125°C	MC68HC805B6FN MC68HC805B6CFN MC68HC805B6MFN

PIN ASSIGNMENTS



VRL
NC
PD5/AN5
PD6/AN6
PD7/AN7
TCMP1
TCMP2
TDO
SCLK
RDI
PC0
PC1
PC2/ECLK
7
52
47
1
VRH
8
PD4/AN4
VDD
PD3/AN3
PD2/AN2
PD1/AN1
PD0/AN0
14
VPP6
OSC1
OSC2
RESET
IRQ
PLMA D/A
20
46
PC3
PC4
PC5
PC6
PC7
VSS
40
VPP1
PB0
PB1
PB2
PB3
PB4
34
PB5
21
27
33
PLMBD/A
TCAP1
TCAP2
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PB7
PB6

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC805C4

Technical Summary

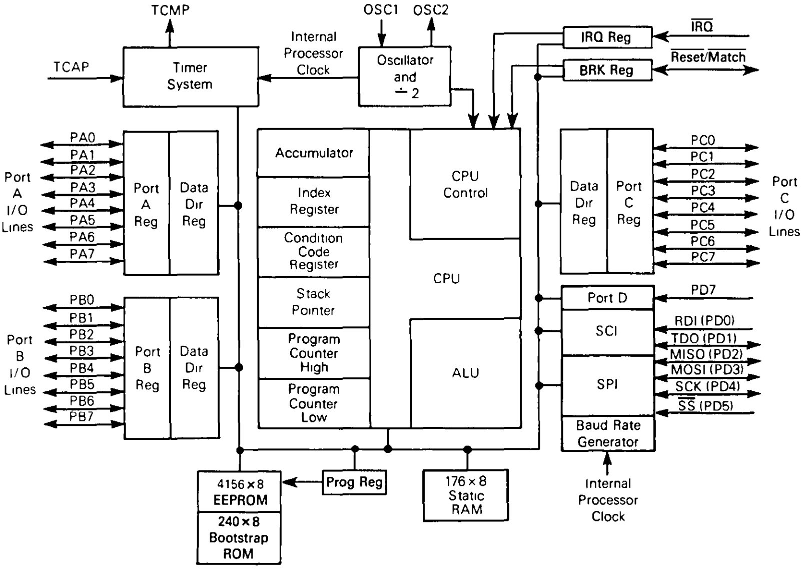
8-Bit Microcontroller Unit

The MC68HC805C4 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 4156 Bytes of On-Chip EEPROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- On-Chip Bootstrap Firmware
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5 5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8×8 Unsigned Multiply Instruction

BLOCK DIAGRAM


TCMP
TCAP
Timer System
Internal Processor Clock
OSC1
OSC2
Oscillator and ÷ 2
IRQ Reg
BRK Reg
IRQ
Reset/Match
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I/O Lines
Port A Reg
Data Dir Reg
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I/O Lines
Port B Reg
Data Dir Reg
Accumulator
Index Register
Condition Code Register
Stack Pointer
Program Counter High
Program Counter Low
CPU Control
CPU
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
Port D
PD7
SCI
RDI (PD0)
TDO (PD1)
SPI
MISO (PD2)
MOSI (PD3)
SCK (PD4)
SS (PD5)
Baud Rate Generator
Internal Processor Clock
4156×8 EEPROM
240×8 Bootstrap ROM
Prog Reg
176×8 Static RAM

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{OSC} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

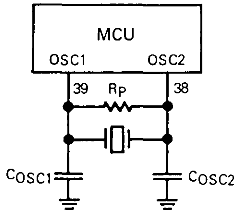
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

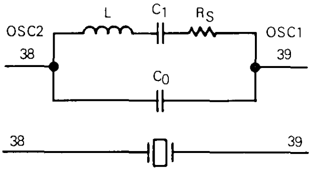
Ceramic Resonator

	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	–

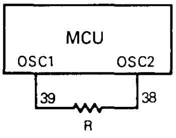
(a) Crystal/Ceramic Resonator Parameters



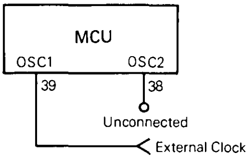
(b) Crystal/Ceramic Resonator Oscillator Connections



(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections



(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

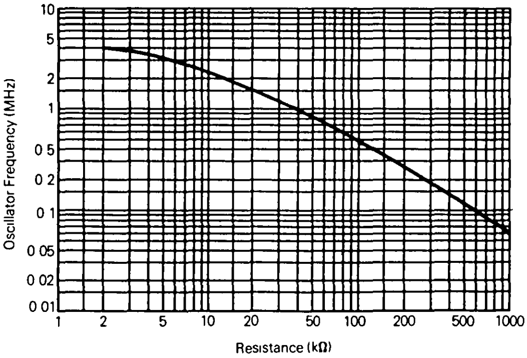


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}/\overline{\text{MATCH}}$

This pin is used as an input to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low. As an output, $\overline{\text{MATCH}}$ generates an address match breakpoint pulse. Refer to **HARDWARE BREAKPOINT REGISTERS** for more information.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

V_{PP}

This pin supplies high voltage to the MCU for programming and erasing the on-chip EEPROM. Refer to **Electrical Specifications** for Vpp specifications.

PROGRAMMING

Input/output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I/O pin functions.

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{\text{W}}$ is an internal signal.

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

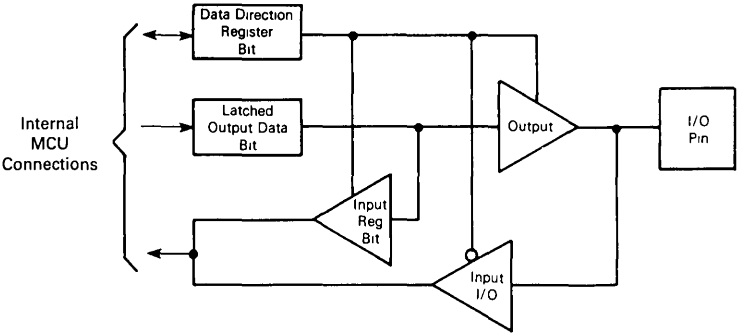


Figure 3. Typical Port I/O Circuit

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of EEPROM, user RAM, bootstrap ROM, control registers, and I/O. The user defined reset and interrupt vectors are located from $1FF4 to $1FFF and are implemented in EEPROM.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

EEPROM

The MCU has 4144 bytes of user EEPROM and 12 bytes of user vector EEPROM. The EEPROM is divided into two arrays, designated A and B. Array A consists of page zero EEPROM ($0020 to $004F) and locations $0100 to $09FF. Array B consists of locations $0A00 to $10FF and locations $1FF0 to $1FFF. Each array has assigned programming address and data buses, which are latched while a programming function is being performed. Separate arrays allow program execution in one array while the other array is being programmed or erased.

Either single- or multi-byte EEPROM programming can be performed. Single-byte programming uses the program register at location $001C. Multi-byte programming uses the bootstrap mode operation. Both types of programming are described in the following paragraphs.

Erasing

Array A and array B can be erased independently or simultaneously. Figure 5 illustrates the erasing operation. Both arrays can be erased simultaneously by setting the LATA and LATB bits, writing to a byte in both arrays, and setting the EEPGM bit.

Single-Byte Programming

The program register ($1C) is used for single-byte EEPROM programming.

	7	6	5	4	3	2	1	0
	0	CPEN	0	0	ERASE	LATA	LATB	EEPGM
RESET	0	0	0	0	0	0	0	0

CPEN — Charge Pump Enable
1 = Charge pump enabled
0 = Charge pump disabled

ERASE — Erase EEPROM Enable
1 = Erase enabled
0 = Erase disabled

LATA — Latch A Enable
1 = Enables array A data and address bus latches for programming or erasing on the next byte write cycle
0 = Latch disabled

LATB — Latch B Enable
1 = Enables array B data and address bus latches for programming or erasing on the next byte write cycle
0 = Latch disabled

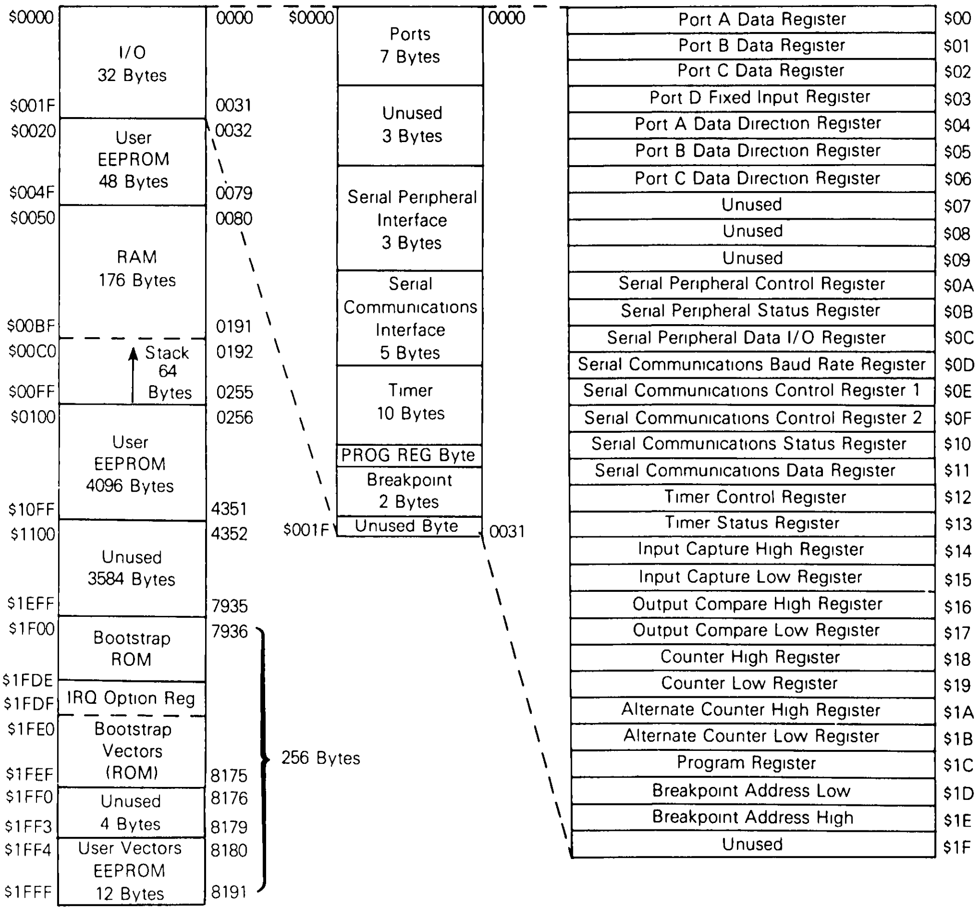

$0000
I/O
32 Bytes
$001F
$0020
User
EEPROM
48 Bytes
$004F
$0050
RAM
176 Bytes
$00BF
$00C0
Stack
64
Bytes
$00FF
$0100
User
EEPROM
4096 Bytes
$10FF
$1100
Unused
3584 Bytes
$1EFF
$1F00
Bootstrap
ROM
$1FDE
$1FDF
IRQ Option Reg
$1FE0
Bootstrap
Vectors
(ROM)
$1FEF
$1FF0
Unused
4 Bytes
$1FF3
$1FF4
User Vectors
EEPROM
12 Bytes
$1FFF
0000
0031
0032
0079
0080
0191
0192
0255
0256
4351
4352
7935
7936
8175
8176
8179
8180
8191
256 Bytes
$0000
Ports
7 Bytes
Unused
3 Bytes
Serial Peripheral
Interface
3 Bytes
Serial
Communications
Interface
5 Bytes
Timer
10 Bytes
PROG REG Byte
Breakpoint
2 Bytes
$001F
Unused Byte
0000
0031
Port A Data Register
Port B Data Register
Port C Data Register
Port D Fixed Input Register
Port A Data Direction Register
Port B Data Direction Register
Port C Data Direction Register
Unused
Unused
Unused
Serial Peripheral Control Register
Serial Peripheral Status Register
Serial Peripheral Data I/O Register
Serial Communications Baud Rate Register
Serial Communications Control Register 1
Serial Communications Control Register 2
Serial Communications Status Register
Serial Communications Data Register
Timer Control Register
Timer Status Register
Input Capture High Register
Input Capture Low Register
Output Compare High Register
Output Compare Low Register
Counter High Register
Counter Low Register
Alternate Counter High Register
Alternate Counter Low Register
Program Register
Breakpoint Address Low
Breakpoint Address High
Unused
$00
$01
$02
$03
$04
$05
$06
$07
$08
$09
$0A
$0B
$0C
$0D
$0E
$0F
$10
$11
$12
$13
$14
$15
$16
$17
$18
$19
$1A
$1B
$1C
$1D
$1E
$1F

Figure 4. Memory Map

EEPGM — Electrically Erase/Program

1 = Applies Vpp power to the EEPROM array for programming or erasing operation.

0 = Vpp power off

If LATA and LATB are cleared, EEPGM cannot be set.

Bits 4, 5, 7 — Not used, always read zero

Figure 6 illustrates the EEPROM single-byte programming operation.

Multi-Byte Programming

The multi-byte EEPROM programming technique can be used to load a user program into the EEPROM to emulate the MC68HC05C4 device. A user program contained in EPROM can be copied into the MC68HC805C4 internal EEPROM.

The MCU device is inserted into the circuit shown in Figure 7. A programming routine is selected via switches S1 through S4, and V_{DD} and Vpp applied to the circuit. Switch S5 changes the MCU from RESET to RUN mode, contol transfers to the bootstrap ROM, and the selected routine is executed.

The EEPROM programming sequence of events is as follows:

1. Place S5 in the RESET position.
2. Select routine with S1 through S5.

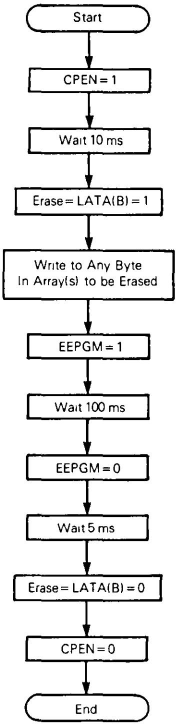

Start
CPEN = 1
Wait 10 ms
Erase = LATA(B) = 1
Write to Any Byte
In Array(s) to be Erased
EEPGM = 1
Wait 100 ms
EEPGM = 0
Wait 5 ms
Erase = LATA(B) = 0
CPEN = 0
End

Figure 5. EEPROM Erasing

3. Apply V_{DD} and V_{PP} to the circuit.
4. Place S5 in the RUN position.
5. Programming routine is executed.
6. Place S5 in the RESET position.
7. Remove V_{DD} and V_{PP}, or select and run new routine.

Once in bootstrap mode, the mode switch settings establish the routine to be executed. The routines are as follows:

Program and verify EEPROM
Bulk erase and verify EEPROM
Load program in RAM and execute
Verify EEPROM contents
Dump EEPROM contents
Execute program in RAM

Program and Verify EEPROM

The program and verify routine copies the contents of an external 8K×8 EPROM into the EEPROM of the MCU, with direct correspondence between the addresses. Memory addresses in the MCU that are not implemented in EEPROM are skipped. Unprogrammed EPROM addresses should contain $FF bytes to speed up the programming process. During programming, the PROGRAMMING LED (DS2) lights. After programming, DS2 turns off and verification begins. If the contents of the external EPROM and MCU internal EEPROM exactly match, the VERIFIED LED (DS1) lights. If a discrepancy is detected, the routine stops and the error address location is placed on the external memory address bus.

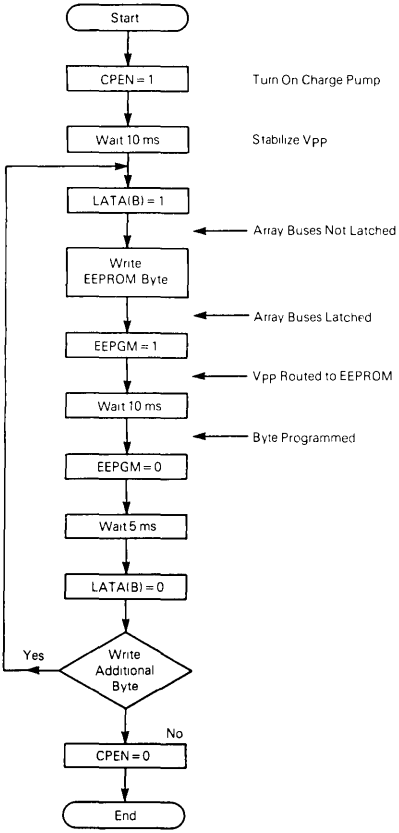

Start
CPEN = 1
Turn On Charge Pump
Wait 10 ms
Stabilize Vpp
LATA(B) = 1
Array Buses Not Latched
Write
EEPROM Byte
Array Buses Latched
EEPGM = 1
Vpp Routed to EEPROM
Wait 10 ms
Byte Programmed
EEPGM = 0
Wait 5 ms
LATA(B) = 0
Yes
Write
Additional
Byte
No
CPEN = 0
End

Figure 6. Single-Byte EEPROM Programming

NOTE

Devices from the A65G mask set do not automatically verify at the end of the programming routine. When programming is complete (DS2 turns off), reset the MCU and perform the procedure to verify EEPROM contents.

Bulk Erase and Verify EEPROM

In the bulk erase and verify EEPROM routine, all EEPROM locations return to the unprogrammed ($FF) state. After the erase operation, every location is verified to be $FF, and the VERIFIED LED (DS1) is lit. If a location fails to erase, the address of the failing location is placed on the external memory address bus. The routine can be executed again until all EEPROM locations are erased and verified.

During this routine, external memory is disabled, port B is set to output data, and $FF data is written to the output register.

Load Program in RAM and Execute

In the load program in RAM and execute routine, user programs are loaded via the SCI port, and then executed. Data is loaded sequentially, starting at address $0050. After the last byte is loaded, control is transferred to the RAM program starting at location $0051. The first byte loaded is the count of the total number of bytes in the program, plus the count byte. The program starts at the second location in RAM. During initialization, the SCI is configured for NRZ data format (idle line, start bit, 8 data bytes, and stop bit). The baud rate is 4800 with a 2-MHz crystal.

Execution can be held off by setting the byte count to a value greater than the number of bytes to be loaded. After loading the last byte, the firmware waits for more data. At this point, S5 can be placed in the RESET position, which resets the MCU with the RAM data intact. All other routines can be entered, including the one to execute program in RAM, by selecting the routine desired and switching S5 to RUN. At the end of the RAM load routine, ports A and C are set to output.

Verify EEPROM Contents

The verify EPROM routine is normally entered automatically after the EEPROM is programmed or erased. Direct entry of this routine causes the EEPROM contents to be compared to the contents of external memory locations at the same addresses. Both DS1 and DS2 are turned off until verification is complete. After verification, DS1 lights. If verification fails, the routine halts with the failing address on the external memory address bus.

Dump EEPROM Contents

In the dump EEPROM contents routine, the EEPROM contents are dumped sequentially to the SCI output. The first location sent is $0020, and the last location set is $1FFF. Unused locations are skipped so that no gaps exist in the data stream. The external memory address lines indicate the current location being sent. Data is sent in NRZ format, as in the load program in RAM routine.

Execute Program in RAM

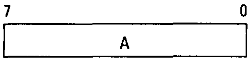
This routine allows the MCU to transfer control to a program previously loaded in RAM. This program is executed once bootstrap mode is entered, if switch S4 is activated, without any firmware initialization. The program must start at location $0051 to be compatible with the load program in RAM routine.

REGISTERS

The MCU contains the registers described in the following paragraphs.

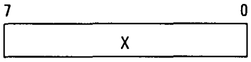
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



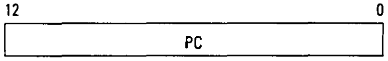
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

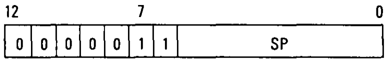
The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

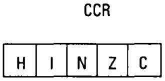
The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack, while an interrupt uses five locations.



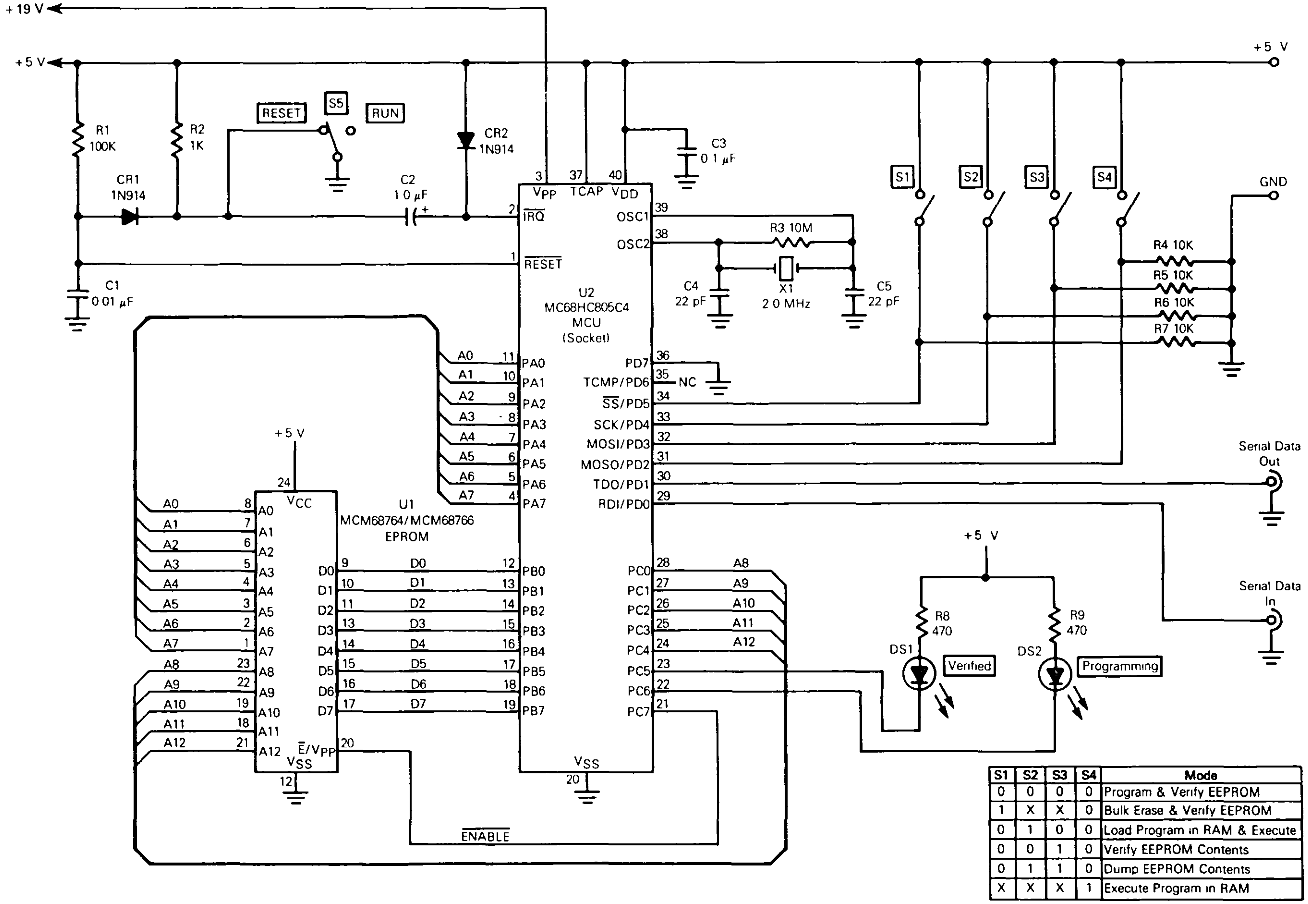
CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.


+19 V
+5 V
+5 V
R1
100K
R2
1K
RESET
S5
RUN
CR2
1N914
C3
0.1 μF
CR1
1N914
C2
1.0 μF
+
3
VPP
37
TCAP
40
VDD
S1
S2
S3
S4
GND
2
IRQ
1
RESET
OSC1
39
OSC2
38
R3 10M
C4
22 pF
X1
2.0 MHz
C5
22 pF
R4 10K
R5 10K
R6 10K
R7 10K
C1
0.01 μF
U2
MC68HC805C4
MCU
(Socket)
A0 11 PA0
A1 10 PA1
A2 9 PA2
A3 8 PA3
A4 7 PA4
A5 6 PA5
A6 5 PA6
A7 4 PA7
36 PD7
35 TCMP/PD6 NC
34 SS/PD5
33 SCK/PD4
32 MOSI/PD3
31 MOSO/PD2
30 TDO/PD1
29 RDI/PD0
Serial Data Out
Serial Data In
+5 V
24
VCC
U1
MCM68764/MCM68766
EPROM
A0 8 A0
A1 7 A1
A2 6 A2
A3 5 A3
A4 4 A4
A5 3 A5
A6 2 A6
A7 1 A7
A8 23 A8
A9 22 A9
A10 19 A10
A11 18 A11
A12 21 A12
D0 9 D0 12 PB0
D1 10 D1 13 PB1
D2 11 D2 14 PB2
D3 13 D3 15 PB3
D4 14 D4 16 PB4
D5 15 D5 17 PB5
D6 16 D6 18 PB6
D7 17 D7 19 PB7
E/VPP 20
VSS 12
PC0 28 A8
PC1 27 A9
PC2 26 A10
PC3 25 A11
PC4 24 A12
PC5 23
PC6 22
PC7 21
+5 V
R8
470
R9
470
DS1
Verified
DS2
Programming
VSS
20
ENABLE

S1	S2	S3	S4	Mode
0	0	0	0	Program & Verify EEPROM
1	X	X	0	Bulk Erase & Verify EEPROM
0	1	0	0	Load Program in RAM & Execute
0	0	1	0	Verify EEPROM Contents
0	1	1	0	Dump EEPROM Contents
X	X	X	1	Execute Program in RAM

Figure 7. EEPROM Programming Circuit

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, and during shifts and rotates

Hardware Breakpoint Register Low (ARL) $1D

	7	6	5	4	3	2	1	0
	A7	A6	A5	A4	A3	A2	A1	A0
RESET	0	0	0	0	0	0	0	0

Hardware Breakpoint Register High (ARH) $1E

	7	6	5	4	3	2	1	0
	0	0	MATCH	A12	A11	A10	A9	A8
RESET	0	0	0	0	0	0	0	0

MATCH — An instruction with the same address as that in the breakpoint register was fetched.
1 = Breakpoint enabled
0 = Breakpoint disabled
A12–A8 — Breakpoint address bits A12 through A8

The hardware breakpoint registers are used as a program debugging aid. The breakpoint address is written into the registers, which are concatenated to form a complete address. When the processor fetches an instruction with the same address, $\overline{\text{RESET/MATCH}}$ goes low for one-half of a machine cycle, then high for one-half of a machine cycle, to ensure that no false resets are generated on successive cycles. This operation does not alter program flow.

The $\overline{\text{RESET/MATCH}}$ pin requires that the user provide an open-drain device during debugging to avoid conflicts. A maximum load of 100 picofarads is allowed on the $\overline{\text{RESET/MATCH}}$ pin during debugging operations.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 4046 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

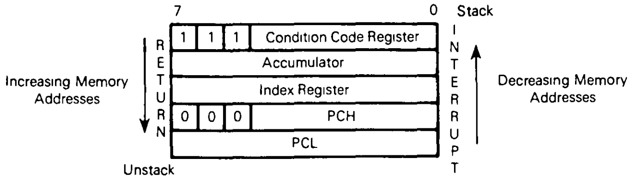
EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts ($\overline{\text{IRQ}}$, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 8.



NOTE Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc Pulling from the stack is in the reverse order

Figure 8. Interrupt Stacking Order

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 9 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at IRQ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available. Sensitivity is controlled by the IRQ option register, described in the following paragraphs. Figure 10 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

IRQ Option Register, $1FDF

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	IRQ	0
RESET	0	0	0	0	0	0	1	0

Bits 7–2, 0 — Not used
Always read zero
IRQ — Interrupt Request Bit Sensitivity
1 = IRQ pin is both negative edge- and level-sensitive
0 = IRQ pin is negative edge-sensitive only
IRQ is set only by reset, but can be cleared by software

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 11).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI During STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that

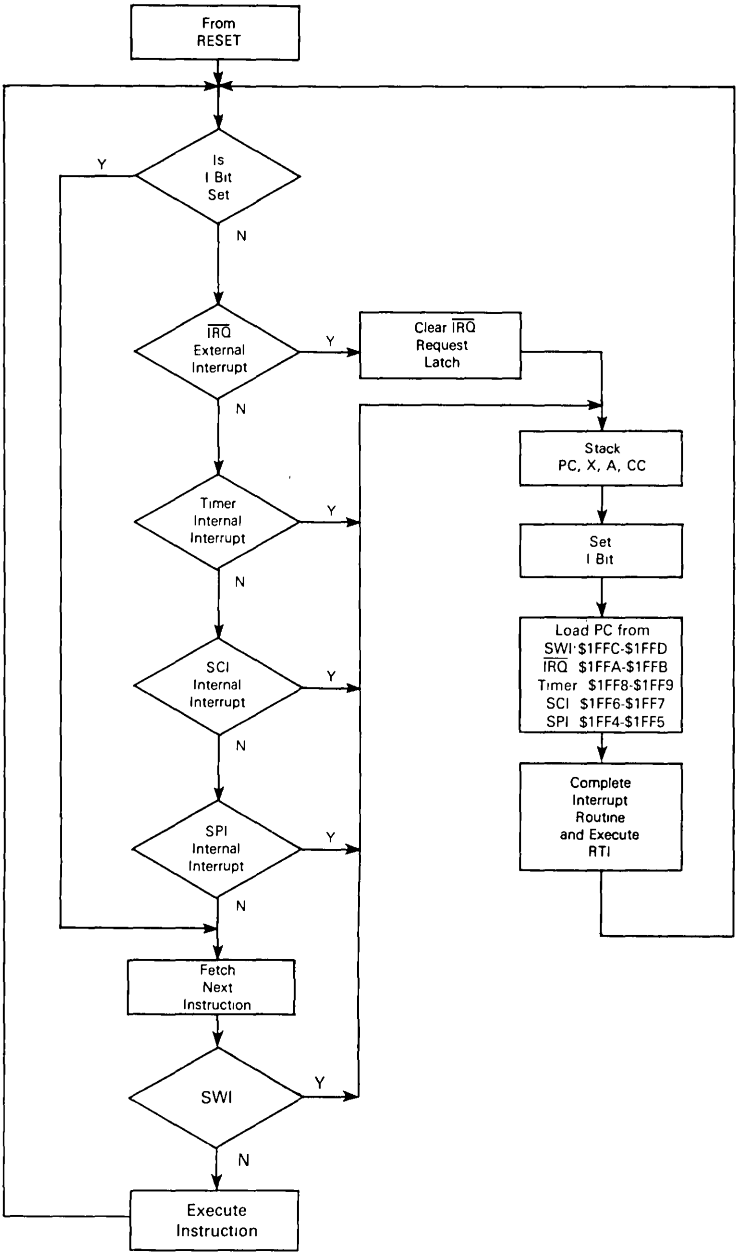
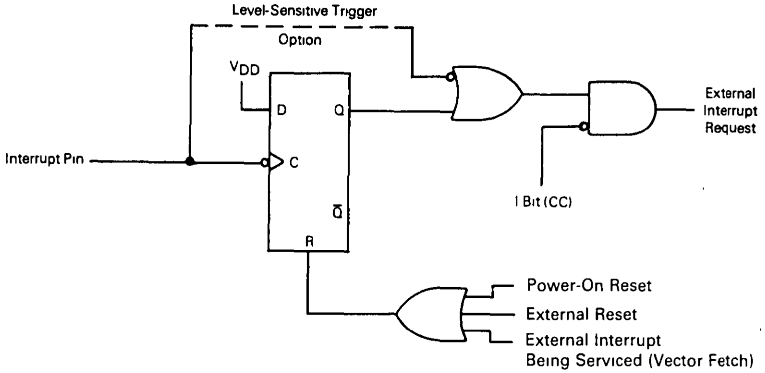
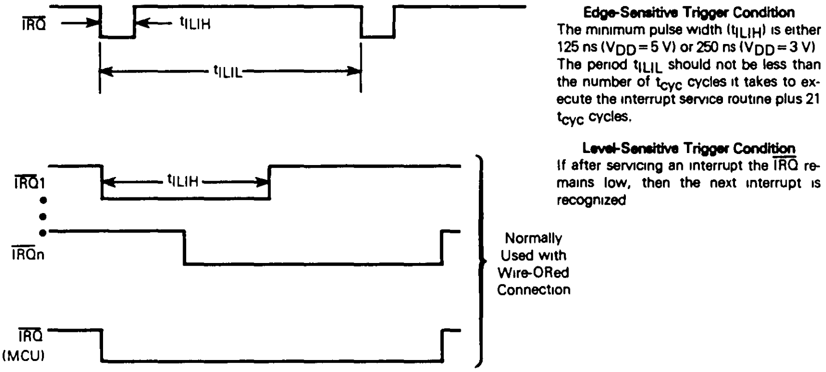

From RESET
Is I Bit Set
Y
N
IRQ External Interrupt
Y
Clear IRQ Request Latch
N
Timer Internal Interrupt
Y
N
Stack PC, X, A, CC
Set I Bit
SCI Internal Interrupt
Y
N
Load PC from
SWI $1FFC-$1FFD
IRQ $1FFA-$1FFB
Timer $1FF8-$1FF9
SCI $1FF6-$1FF7
SPI $1FF4-$1FF5
SPI Internal Interrupt
Y
N
Complete Interrupt Routine and Execute RTI
Fetch Next Instruction
SWI
Y
N
Execute Instruction

Figure 9. Reset and Interrupt Processing Flowchart


Level-Sensitive Trigger
Option
VDD
D
Q
C
Q̄
R
Interrupt Pin
External
Interrupt
Request
I Bit (CC)
Power-On Reset
External Reset
External Interrupt
Being Serviced (Vector Fetch)

(a) Interrupt Internal Function Diagram


IRQ
tILIH
tILIL
IRQ1
IRQn
IRQ
(MCU)
Normally
Used with
Wire-ORed
Connection

Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns ($V_{DD}=5$ V) or 250 ns ($V_{DD}=3$ V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles.

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized

(b) Interrupt Mode Diagram

Figure 10. External Interrupt

transfer is halted. If a low input to the $\overline{IRQ}$ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI During STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the $\overline{IRQ}$ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the $\overline{IRQ}$ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active; refer to Figure 12. An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{\text{RESET}}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 13 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two

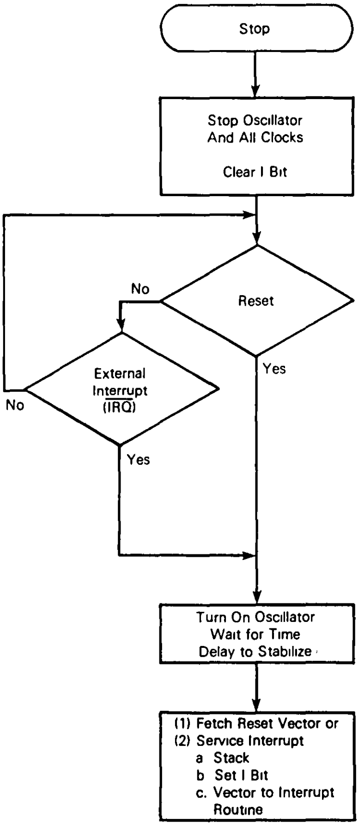

Stop
Stop Oscillator And All Clocks
Clear I Bit
Reset
No
Yes
External Interrupt ($\overline{\text{IRQ}}$)
No
Yes
Turn On Oscillator Wait for Time Delay to Stabilize
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c. Vector to Interrupt Routine

Figure 11. STOP Function Flowchart

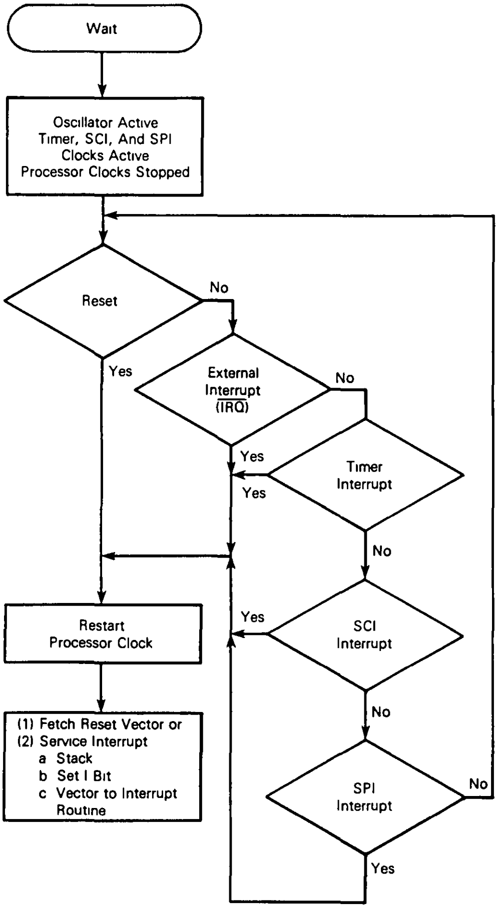

Wait
Oscillator Active
Timer, SCI, And SPI
Clocks Active
Processor Clocks Stopped
Reset
No
Yes
External
Interrupt
(IRQ)
No
Yes
Timer
Interrupt
Yes
No
Restart
Processor Clock
Yes
SCI
Interrupt
No
(1) Fetch Reset Vector or
(2) Service Interrupt
a Stack
b Set I Bit
c Vector to Interrupt
Routine
SPI
Interrupt
No
Yes

Figure 12. WAIT Function Flowchart

registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler which divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations \$18–\$19 (counter register) or \$1A–\$1B (counter alternate register). A read from only

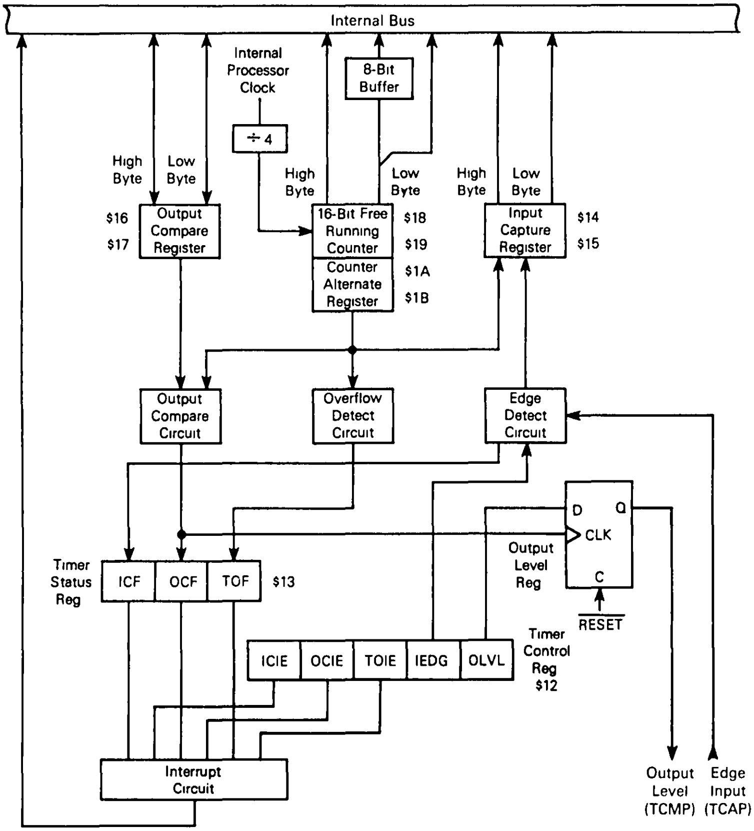

Internal Bus
Internal Processor Clock
8-Bit Buffer
÷ 4
High Byte
Low Byte
High Byte
Low Byte
High Byte
Low Byte
$16
$17
Output Compare Register
16-Bit Free Running Counter
$18
$19
Counter Alternate Register
$1A
$1B
Input Capture Register
$14
$15
Output Compare Circuit
Overflow Detect Circuit
Edge Detect Circuit
D
Q
CLK
C
Output Level Reg
RESET
Timer Status Reg
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
Timer Control Reg $12
Interrupt Circuit
Output Level (TCMP)
Edge Input (TCAP)

Figure 13. Timer Block Diagram

the least significant byte (LSB) of the free running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therfore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC, and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers which make up the 16-bit input capture register are read-only and are used to latch the value of the free running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free running counter, which is four internal bus clock cycles.

The free running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free running counter value which corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

IEDG — Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free running counter transfer to the input capture register
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG bit (U = unaffected).

OLVL — Output Level
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
1 = High output
0 = Low output

Bits 2, 3, and 4 — Not used
Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read only register containing three status flag bits.

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

ICF — Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF — Output Compare Flag
1 = Flag set when output compare register contents match the free running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
1 = Flag set when free running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If reset is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If reset is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software selectable word length (eight or nine bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data which is presented between the internal data bus and the output pin (TDO), and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 14.

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and

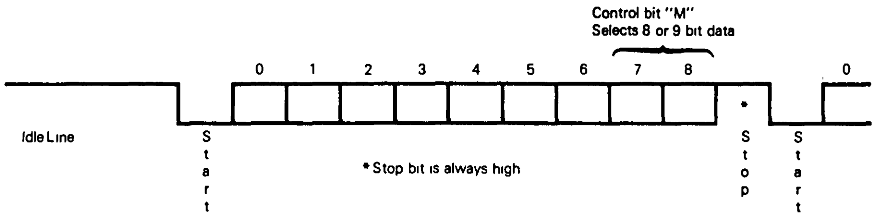


Figure 14. Data Format

interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate which is 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 15); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data which is presented from the transmit data register (TDR), via the SCI, to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 15. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits which individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set. This will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

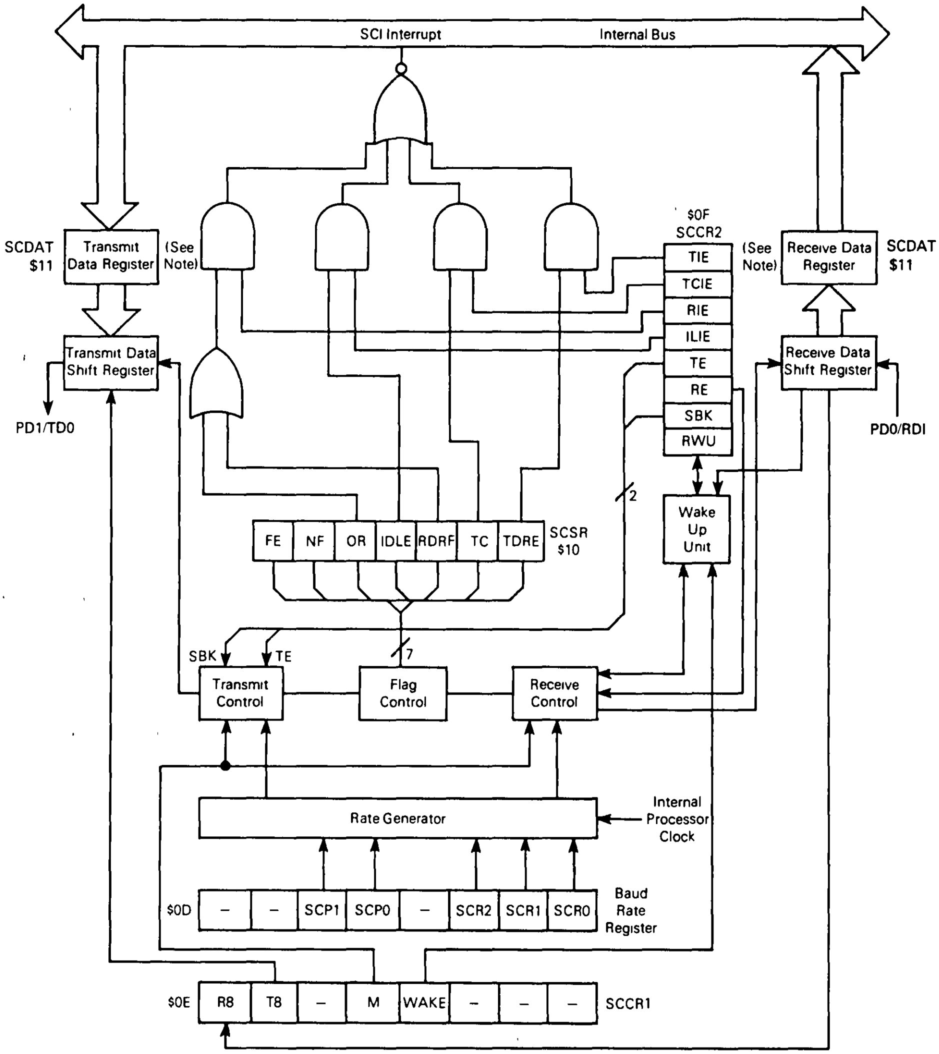
There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read/write register used to receive and transmit SCI data.

	7	6	5	4	3	2	1	0
	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET	U	U	U	U	U	U	U	U

As shown in Figure 15, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.


SCI Interrupt
Internal Bus
SCDAT $11
Transmit Data Register
(See Note)
Transmit Data Shift Register
PD1/TDO
$0F SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(See Note)
Receive Data Register
SCDAT $11
Receive Data Shift Register
PD0/RDI
Wake Up Unit
2
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $10
7
SBK
TE
Transmit Control
Flag Control
Receive Control
Rate Generator
Internal Processor Clock
$0D
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
Baud Rate Register
$0E
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal It is the transmit data register when written and receive data register when read

Figure 15. SCI Block Diagram

Serial Communications Control Register 1 (SCCR1) $0E

The SCCR1 provides control bits which determine word length and select the wake-up method.

7	6	5	4	3	2	1	0
R8	T8	—	M	WAKE	—	—	—
RESET							
U	U	—	U	U	—	—	—

R8 — Receive Data Bit 8
R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).
T8 — Transmit Data Bit 8
T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).
M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit
WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most-significant bit)
0 = Idle line condition
Bits 0–2, and 5 — Not used
Can read either one or zero

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $0F

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET.							
0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt enabled
0 = TDRE interrupt disabled
TCIE — Transmit Complete Interrupt Enable
1 = SCI interrupt enabled
0 = TC interrupt disabled
RIE — Receive Interrupt Enable
1 = SCI interrupt enabled
0 = RDRF and OR interrupts disabled
ILIE — Idle Line Interrupt Enable
1 = SCI interrupt enabled
0 = Idle interrupt disabled
TE — Transmit Enable
1 = Transmit shift register output applied to the TD0 line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
0 = Transmitter disabled after last byte loaded in the SCDAT and TDRE set. After last byte is transmitted, TD0 line becomes a high-impedance line.
RE — Receive Enable
1 = Receiver shift register input applied to the RDI line
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits inhibited
RWU — Receiver Wake Up
1 = Places receiver in sleep mode and enables wake-up function
0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)
SBK — Send Break
1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET							
1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
1 = TDR contents transferred to the transmit data shift register
0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1) followed by a write to the TDR.
TC — Transmit Complete
1 = Indicates end of data frame, preamble, or break condition has occurred
0 = TC bit cleared by reading the SCSR (with TC = 1) followed by a write to the TDR

RDRF — Receive Data Register (RDR) Full

1 = Receiver data shift register contents transferred to the RDR

0 = Receiver data shift register transfer did not occur. RDRF cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR

IDLE — Idle Line Detect

1 = Indicates receiver has detected an idle line

0 = IDLE cleared by reading the SCSR (with IDLE = 1) followed by a read of the RDR. Once cleared, IDLE cannot be set until RDI line becomes active and idle again.

OR — Overrun Error

1 = Indicates receive data shift register data sent to a full RDR (RDRF = 1). Data causing the overrun is lost and RDR data is not disturbed.

0 = OR cleared by reading the SCSR (with OR = 1) followed by a read of the RDR

NF — Noise Flag

1 = Indicates noise present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.

0 = NF cleared by reading the SCSR (with NF = 1) followed by a read of the RDR

FE — Framing Error

1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.

0 = NF cleared by reading the SCSR (with FE = 1) followed by a read of the RDR

Bit 0 — Not used

Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

7	6	5	4	3	2	1	0
—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET:							
—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0

SCP1 — SCI Prescaler Bit 1

Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 2.

SCR0 — SCI Baud Rate Bit 0

SCR1 — SCI Baud Rate Bit 1

SCR2 — SCI Baud Rate Bit 2

Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 3.

Tables 2 and 3 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 2 represent the final baud rate resulting from the internal processor clock division shown in the divided by column only (prescaler division only). Table 3 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600 Hz baud rate is required with a 2.4576 MHz external

Table 2. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	0		4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	10 082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 2 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 3. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE. Table 3 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receive clock is 16 times higher in frequency than the actual baud rate.

crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 16) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programamble master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most-significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}=1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 17, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{SS}$) input line selects a slave device. The $\overline{SS}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high; if the $\overline{SS}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{SS}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

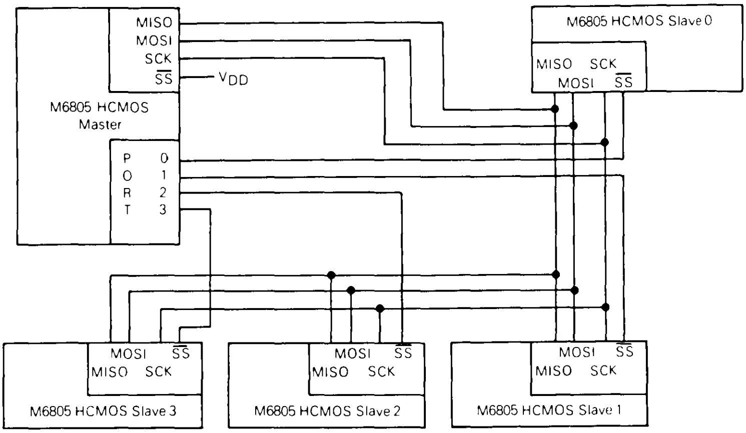


Figure 16. Master-Slave System Configuration

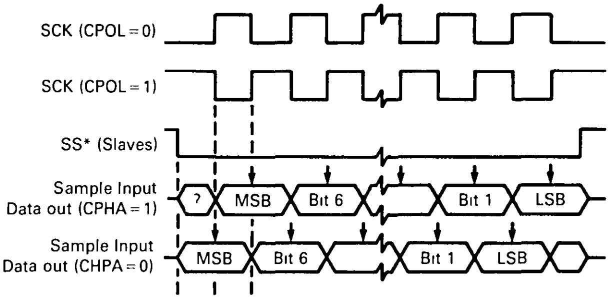

SCK (CPOL=0)
SCK (CPOL=1)
SS* (Slaves)
Sample Input
Data out (CPHA=1)
?
MSB
Bit 6
Bit 1
LSB
Sample Input
Data out (CHPA=0)
MSB
Bit 6
Bit 1
LSB

Figure 17. Data Clock Timing Diagram

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 18. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 19 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

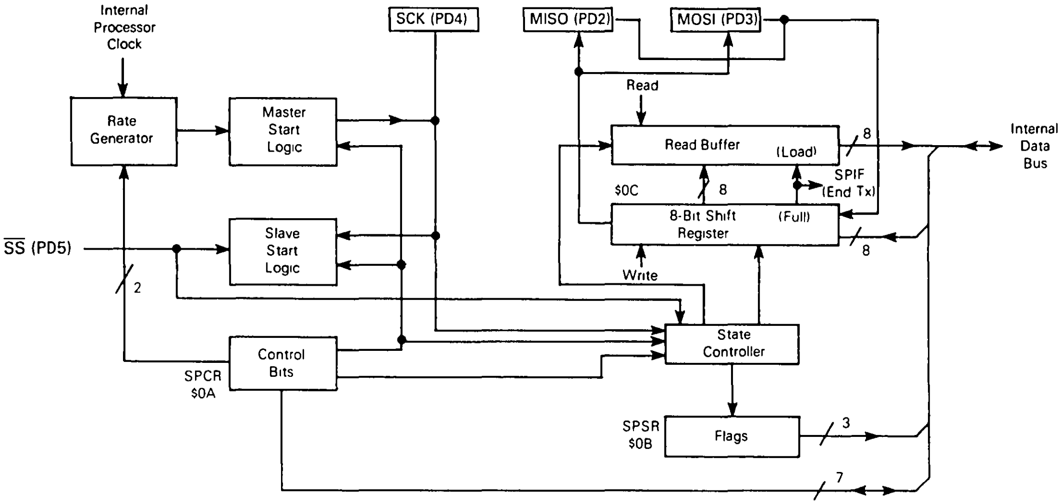

Internal Processor Clock
Rate Generator
Master Start Logic
SCK (PD4)
MISO (PD2)
MOSI (PD3)
Read
Read Buffer
(Load)
8
Internal Data Bus
$0C
8
SPIF
(End Tx)
8-Bit Shift Register
(Full)
8
SS (PD5)
Slave Start Logic
Write
2
State Controller
Control Bits
SPCR
$0A
SPSR
$0B
Flags
3
7

Figure 18. SPI Block Diagram

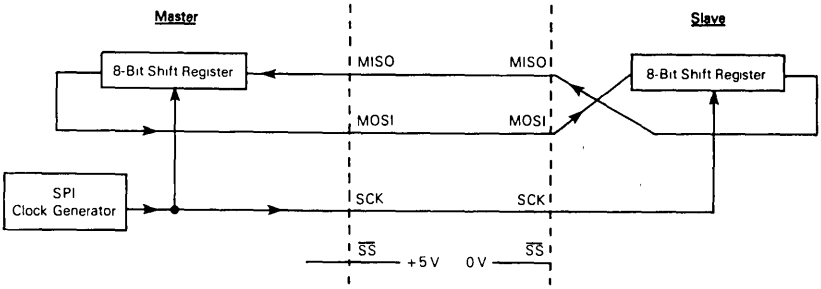


Figure 19. SPI Master-Slave Interconnections

Serial Peripheral Control Register (SPCR) $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

7	6	5	4	3	2	1	0
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:							
0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase
Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits
Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. If in the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 — Not used
Can read either one or zero

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR) $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL	—	MODF	—	—	—	—
RESET							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device
(If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing accomplished by reading SPSR (with SPIF = 1) followed by SPDR access

WCOL — Write Collision
1 = Indicates an attempt made to write to SPDR while data transfer is in process
0 = Clearing accomplished by reading SPSR (with WCOL = 1) followed by SPDR access

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict
0 = Clearing accomplished by reading SPSR (with MODF = 1) followed by a write to the SPCR

Bits 0–3, and 5 — Not used
Can read either zero or one

Serial Peripheral Data I/O Register (SPDR) $0C

The SPDR is a read/write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

<table>
<tr><td>Operation</td><td colspan="4">X:A X*A</td></tr>
<tr><td>Description</td><td colspan="4">Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register</td></tr>
<tr><td>Condition Codes</td><td colspan="4">H: Cleared
I: Not affected
N: Not affected
Z: Not affected
C: Cleared</td></tr>
<tr><td>Source</td><td colspan="4">MUL</td></tr>
<tr><td rowspan="2">Form(s)</td><td>Addressing Mode</td><td>Cycles</td><td>Bytes</td><td>Opcode</td></tr>
<tr><td>Inherent</td><td>11</td><td>1</td><td>$42</td></tr>
</table>

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 4 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the

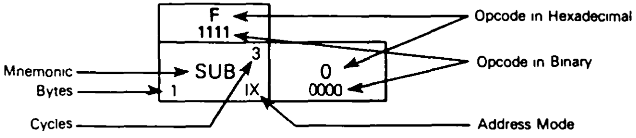
Table 4. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to 2×V_{DD}+0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC805C4 (Standard) MC68HC805C4 (Extended)	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C
Programming Voltage	V_{PP}	V_{DD} −0.3 to 20	V

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

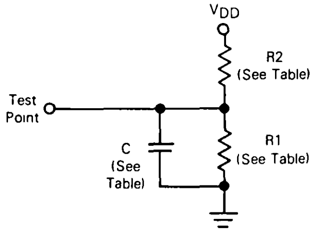
Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		60	
Plastic Leaded Chip Carrier (PLCC)		70	

$V_{DD} = 4.5$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0, PD5, PD7	1.9 kΩ	2.26 kΩ	200 pF

$V_{DD} = 3.0$ V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ	6.32 kΩ	50 pF
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF



3

Figure 20. Equivalent Test Load

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected. The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 21) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 22)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 23) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 24 and 25) Wait (see Figures 24 and 25) Stop (see Figure 25) 25°C 0° to 70°C (Standard) −40° to +85°C	I_{DD}	 — — — — —	 4.5 1.9 2 0 — —	 8 0 4.3 50 140 180	 mA mA μA μA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. An extended temperature (−40° to +85°C) version and a 25°C only version are available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage	V_{OH}				V
(I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 21)		$V_{DD} - 0.3$	—	—	
(I_{Load} = 0.4 mA) PD1-PD4 (see Figure 22)		$V_{DD} - 0.3$	—	—	
Output Low Voltage (see Figure 23) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 24 and 26)		—	1.0	2.5	mA
Wait (see Figures 24 and 26)		—	0.5	1.4	mA
Stop (see Figure 26)					
25°C		—	1.0	30	µA
0° to 70°C (Standard)		—	—	80	µA
−40° to +85°C		—	—	120	µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, PD0-PD5, PD7	C_{in}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. An extended temperature (−40° to +85°C) version and a 25°C only version are available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 0.5 Vdc%, V_{SS} = 0 Vdc)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage/Erase Voltage	V_{PP}	18.5	19	19.5	V
V_{PP} Supply Current	I_{PP}				
$V_{PP} = V_{DD}$		—	1	10	µA
V_{PP} = Programming Voltage		—	TBD	TBD	µA

NOTE:

V_{PP} should always be connected to V_{DD} except during programming/erasing.

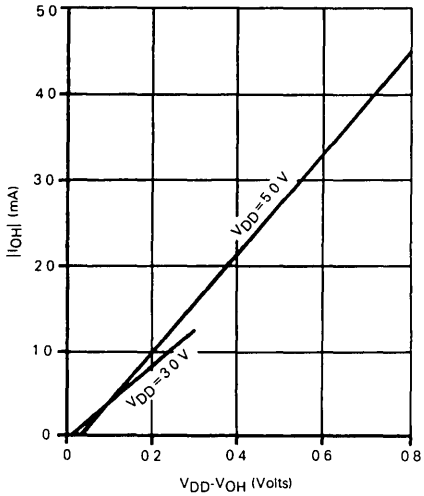


Figure 21. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

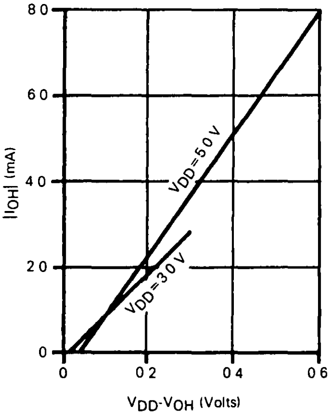


Figure 22. Typical V_{OH} vs I_{OH} for PD1-PD4

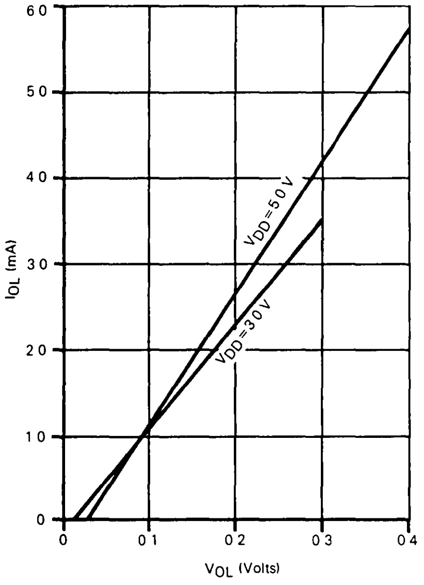


Figure 23. Typical V_{OL} vs I_{OL} for all Ports

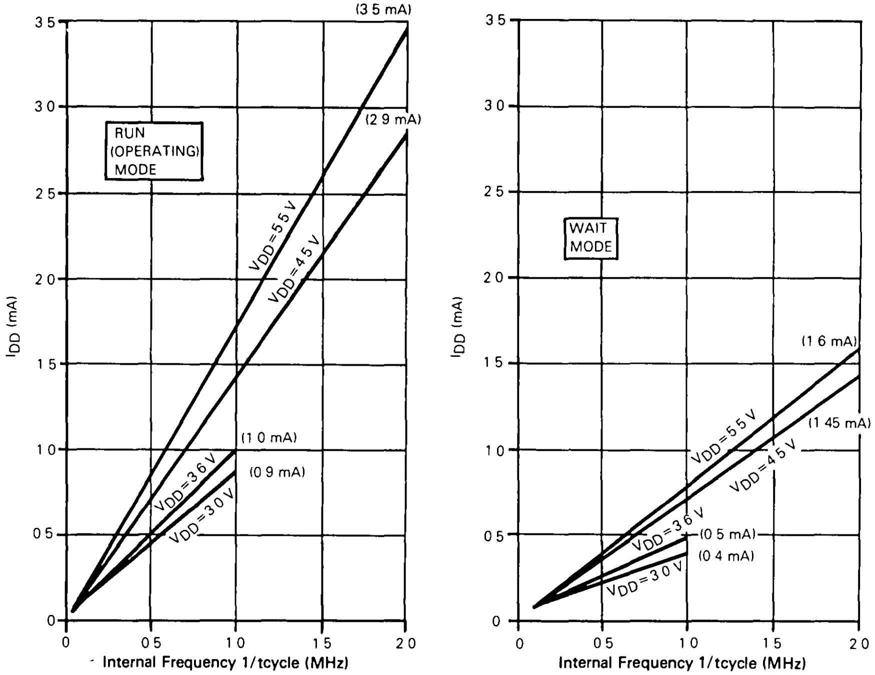


Figure 24. Typical Current vs Internal Frequency for Run and Wait Modes

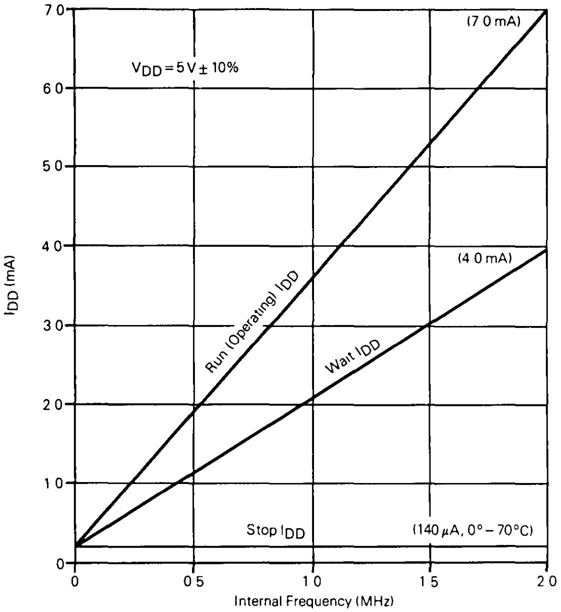


Figure 25. Maximum I_{DD} vs Frequency for V_{DD} = 5.0 Vdc

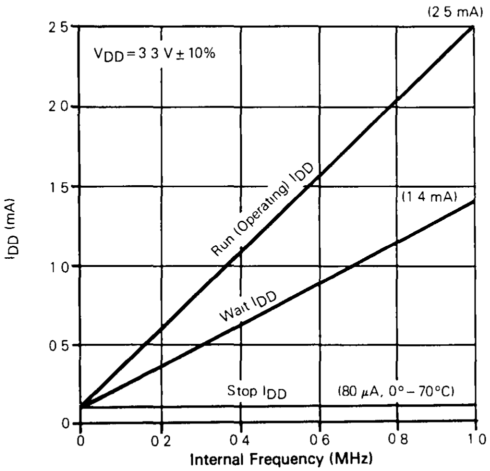


Figure 26. Maximum I_{DD} vs Frequency for V_{DD} = 3.3 Vdc

CONTROL TIMING

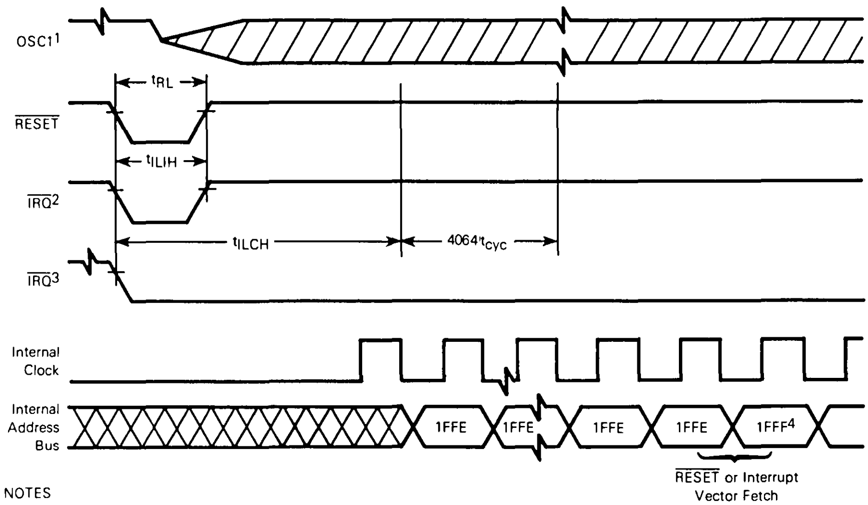
($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 4.2 4.2	MHz
Internal Operating Frequency Crystal ($f_{osc}-2$) External Clock ($f_{osc}-2$)	f_{op}	 — dc	 2.1 2.1	MHz
Cycle Time (see Figure 30)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 30)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 27)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (see Figure 30)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 28) Input Capture Pulse Period (see Figure 28)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 30)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 30)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES
1 Represents the internal gating of the OSC1 pin
2 $\overline{\text{IRQ}}$ pin edge-sensitive mask option
3 $\overline{\text{IRQ}}$ pin level and edge-sensitive mask option
4 $\overline{\text{RESET}}$ vector address shown for timing example

Figure 27. Stop Recovery Timing Diagram

CONTROL TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}			MHz
Crystal Option		—	2.0	
External Clock Option		dc	2.0	
Internal Operating Frequency	f_{op}			MHz
Crystal ($f_{osc} \div 2$)		—	1.0	
External Clock ($f_{osc} \div 2$)		dc	1.0	
Cycle Time (see Figure 30)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 30)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 27)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 30)	t_{RL}	1.5	—	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (see Figure 28)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (see Figure 28)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 30)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 30)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

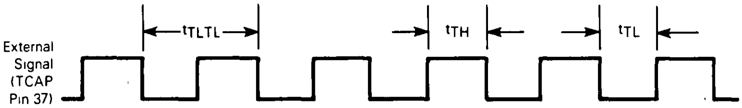


Figure 28. Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 29)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 240	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.

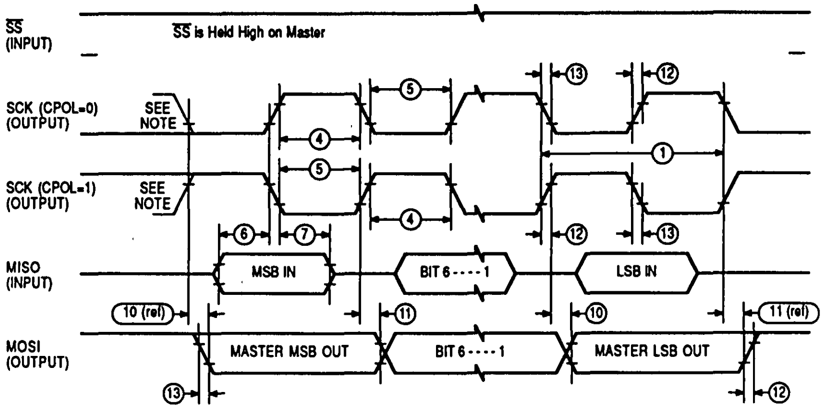
SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 29)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	µs
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	720	—	µs
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	720	—	µs
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	250	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	500	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

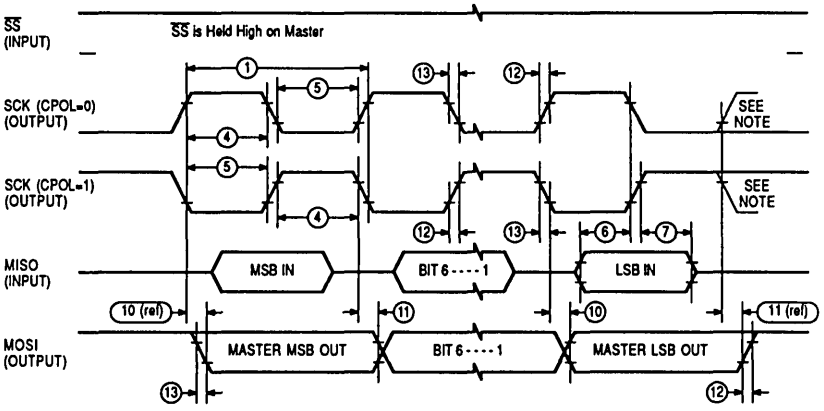
*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.


SS
(INPUT)
SS is Held High on Master
SCK (CPOL=0)
(OUTPUT)
SEE
NOTE
SCK (CPOL=1)
(OUTPUT)
SEE
NOTE
MISO
(INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI
(OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

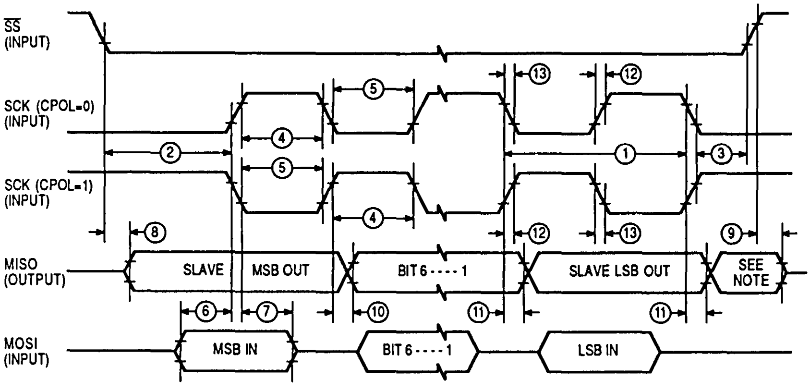
a) SPI MASTER TIMING (CPHA = 0)


SS
(INPUT)
SS is Held High on Master
SCK (CPOL=0)
(OUTPUT)
SEE
NOTE
SCK (CPOL=1)
(OUTPUT)
SEE
NOTE
MISO
(INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI
(OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

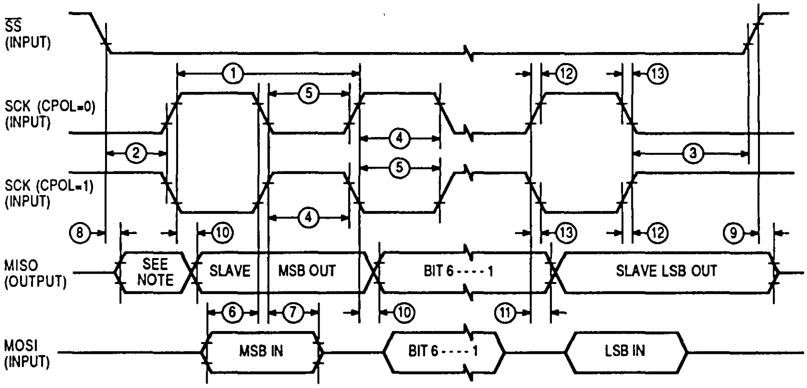
b) SPI MASTER TIMING (CPHA = 1)

Figure 29. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE. Not defined but normally MSB of character just received.

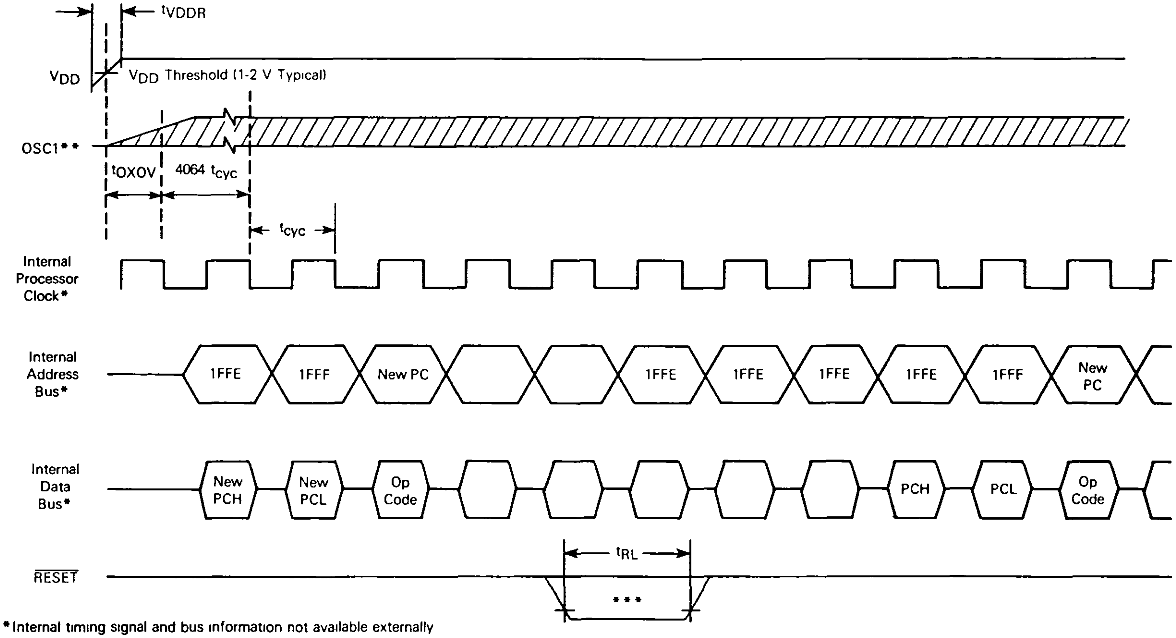
c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE: Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA = 1)

Figure 29. SPI Timing Diagrams (Sheet 2 of 2)


t_{VDDR}
V_{DD}
V_{DD} Threshold (1-2 V Typical)
OSC1**
t_{OXOV}
4064 t_{cyc}
t_{cyc}
Internal Processor Clock*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
RESET
t_{RL}
***

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency It is only used to represent time

***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence

Figure 30. Power-On Reset and $\overline{\text{RESET}}$

MECHANICAL DATA

This section contains ordering informatiom, pin assignments, and package dimensions for the MC68HC805C4.

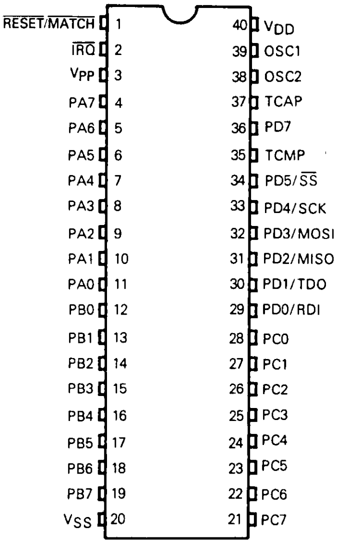
ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC805C4 device.

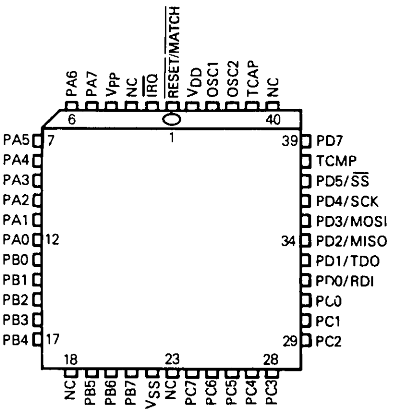
Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0°C to +70°C −40°C to +85°C	MC68HC805C4P MC68HC805C4CP
PLCC (FN Suffix)	0°C to +70°C −40°C to +85°C	MC68HC805C4FN MC68HC805C4CFN

PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE


RESET/MATCH 1
IRQ 2
VPP 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TCAP
36 PD7
35 TCMP
34 PD5/SS
33 PD4/SCK
32 PD3/MOSI
31 PD2/MISO
30 PD1/TDO
29 PD0/RDI
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7

44-LEAD PLCC PACKAGE


PA6
PA7
VPP
NC
IRQ
RESET/MATCH
VDD
OSC1
OSC2
TCAP
NC
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
PB4 17
39 PD7
TCMP
PD5/SS
PD4/SCK
PD3/MOSI
34 PD2/MISO
PD1/TDO
PD0/RDI
PC0
PC1
29 PC2
18
23
28
NC
PB5
PB6
PB7
VSS
NC
PC7
PC6
PC5
PC4
PC3

NOTE: Bulk substrate tied to V_{SS}.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146805E2

Technical Summary

8-Bit Microprocessor Unit

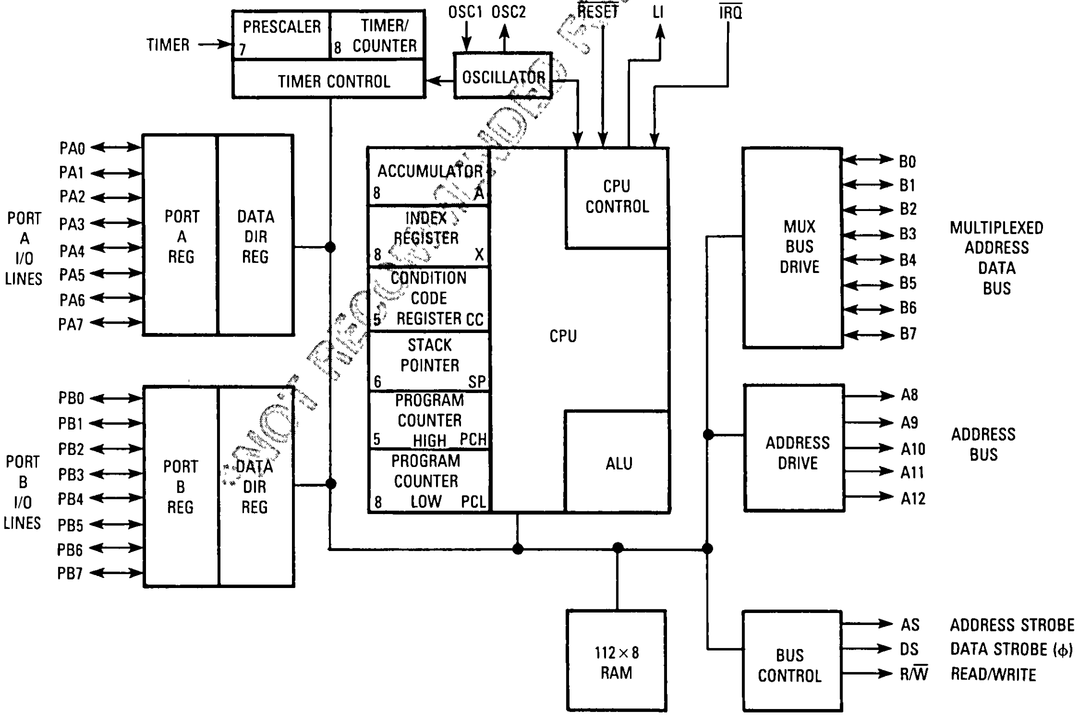
The MC146805E2 (CMOS) Microprocessor Unit (MPU) is a member of the MC146805 Family of microcomputers. This low cost and low power MPU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MPU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual* (M6805UM(AD2)) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MPU.

- Internal 8-Bit Timer with 7-Bit Programmable Prescaler
- On-chip Clock
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Multiplexed Address/Data Bus
- Power-saving STOP and WAIT Modes
- Single 3.0- to 6.0-Volt Supply
- Fully Static Operation

NOT RECOMMENDED FOR NEW DESIGN

BLOCK DIAGRAM


TIMER
PRESCALER
7
TIMER/
8 COUNTER
TIMER CONTROL
OSC1
OSC2
OSCILLATOR
RESET
LI
IRQ
PORT A I/O LINES
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A REG
DATA DIR REG
PORT B I/O LINES
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PORT B REG
DATA DIR REG
ACCUMULATOR
8 A
INDEX REGISTER
8 X
CONDITION CODE REGISTER CC
5
STACK POINTER
6 SP
PROGRAM COUNTER
5 HIGH PCH
PROGRAM COUNTER
8 LOW PCL
CPU CONTROL
CPU
ALU
MUX BUS DRIVE
B0
B1
B2
B3
B4
B5
B6
B7
MULTIPLEXED ADDRESS DATA BUS
ADDRESS DRIVE
A8
A9
A10
A11
A12
ADDRESS BUS
112×8 RAM
BUS CONTROL
AS
DS
R/W
ADDRESS STROBE
DATA STROBE (ϕ)
READ/WRITE

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcomputer using these two pins. V_{DD} is +5 volts (±0.5Δ) power and V_{SS} is ground.

LI (LOAD INSTRUCTION)

The output is used to indicate that a fetch of the next opcode is in progress. The output is used only for certain debugging and test systems. In normal operation, this pin is not connected. This signal overlaps data strobe (DS).

$\overline{\text{IRQ}}$

This pin is a level-sensitive and edge-sensitive input which can be used to request an interrupt sequence. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for the on-chip clock oscillator circuits. A crystal or an external signal is connected to these pins to provide a system clock. Figure 1 shows the crystal connection, and Figure 2 shows OSC1 to bus transitions for system designs using oscillators slower than 5 MHz.

Crystal

The circuit shown in Figure 1 is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

TIMER

This pin is used as an external input to control the internal timer/counter circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program.

$\overline{\text{RESET}}$

This pin has a Schmitt trigger input and an on-chip pullup. The MPU can be reset by pulling $\overline{\text{RESET}}$ low.

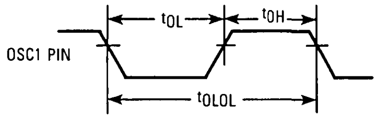
AS (ADDRESS STROBE)

This output strobe is used to indicate the presence of an address on the 8-bit multiplexed bus. The eight least-significant address bits are demultiplexed from the data bus. The output is capable of driving one standard TTL load and 130 picofarads and is available at f_{osc} divided-by-five when the MPU is not in the WAIT or STOP mode.

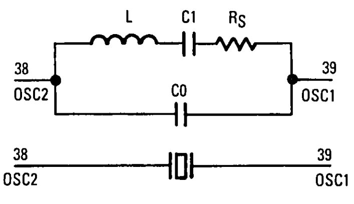
Crystal Parameters Representative Frequencies

	5.0 MHz	4.0 MHz	1.0 MHz
R_S max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0.02 pF	0.012 pF	0.008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-25 pF	15-30 pF

OSCILLATOR WAVEFORM



CRYSTAL CIRCUIT



CRYSTAL OSCILLATOR CONNECTIONS

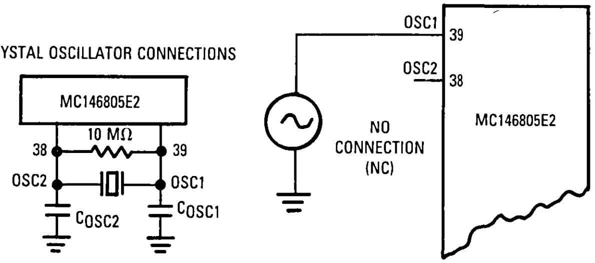
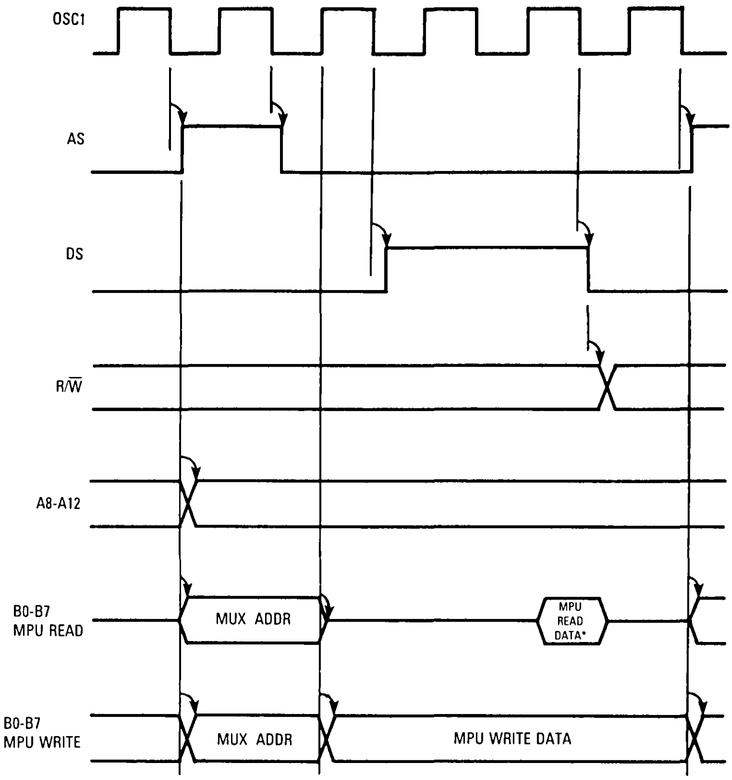


Figure 1. Oscillator Connections



*READ DATA "LATCHED" ON DS FALL

Figure 2. OSC1 to Bus Transitions

DS (DATA STROBE)

This output is used to transfer data to or from a peripheral or memory. DS occurs when the MPU does a data read or write and during data transfer to or from internal memory. The output is capable of driving one standard TTL load and 130 picofarads and is available at f_{OSC} divided-by-five when the MPU is not in the WAIT or STOP mode.

R/W̄ (READ/WRITE)

This output is used to indicate the direction of data transfer to both internal memory and I/O registers, and external peripheral devices and memories. The output indicates to a selected peripheral whether the MPU is to read (R/W̄ high) or write (R/W̄ low) data on the next data strobe. The output is capable of driving one standard TTL load and 130 picofarads.

A8-A12 (HIGH ORDER ADDRESS LINES)

These five output lines constitute the higher order non-multiplexed addresses. Each output line is capable of driving one standard TTL load and 130 picofarads.

B0-B7 (ADDRESS/DATA BUS)

These bidirectional lines constitute the lower order addresses and data. These lines are multiplexed with address present at address strobe time and data present at data strobe time. These lines are bidirectional during data mode as indicated by the R/W̄ pin and are capable of driving one standard TTL load and 130 picofarads.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7)

These 16 lines are arranged into two 8-bit ports. Both ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Any port pin is programmable as either input or output under software control of the corresponding write-only data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output and a logic 0 for input. On reset, all the DDRs are initialized to a logic 0 state to put the ports in the input mode. To avoid undefined levels, the port output registers are not initialized on reset but may be written to before setting the DDR bits.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Care must be exercised when using read-modify-write instructions since the data read corresponds: to the pin level if the DDR is an input (0) and, also, to the latched output when the DDR is an output (1). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

MEMORY

The MPU is capable of addressing 8192 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consist of external memory, user RAM, a timer control register, and I/O. The interrupt vectors are located from $1FF6 to $1FFF. During program reads from on-chip locations, the MPU accepts data only from the addressed on-chip location. Any read data appearing on the input bus is ignored.

Table 1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the shared stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

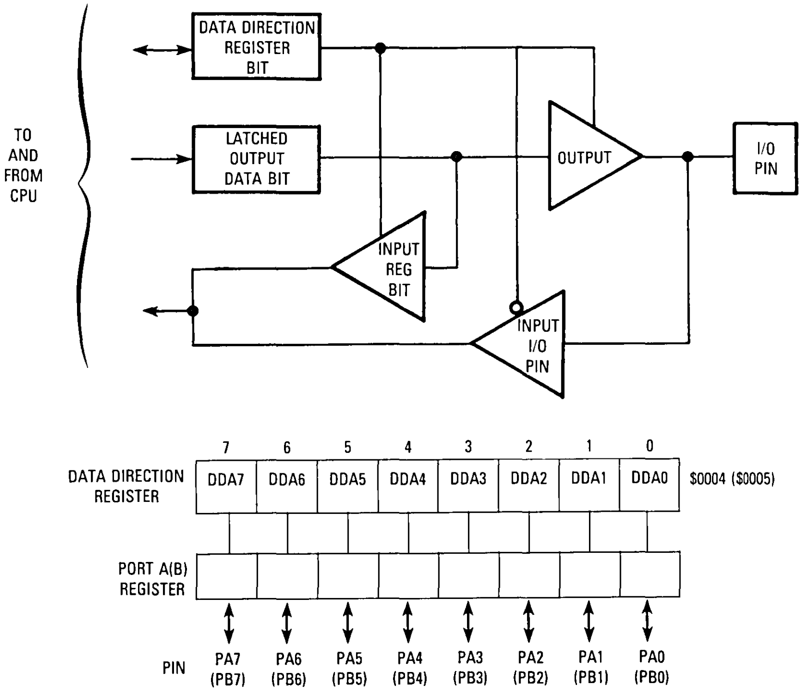

TO
AND
FROM
CPU
DATA DIRECTION REGISTER BIT
LATCHED OUTPUT DATA BIT
OUTPUT
I/O PIN
INPUT REG BIT
INPUT I/O PIN
7
6
5
4
3
2
1
0
DATA DIRECTION REGISTER
DDA7
DDA6
DDA5
DDA4
DDA3
DDA2
DDA1
DDA0
$0004 ($0005)
PORT A(B) REGISTER
PIN
PA7 (PB7)
PA6 (PB6)
PA5 (PB5)
PA4 (PB4)
PA3 (PB3)
PA2 (PB2)
PA1 (PB1)
PA0 (PB0)

Figure 3. Typical Port I/O Circuitry and Register Configuration

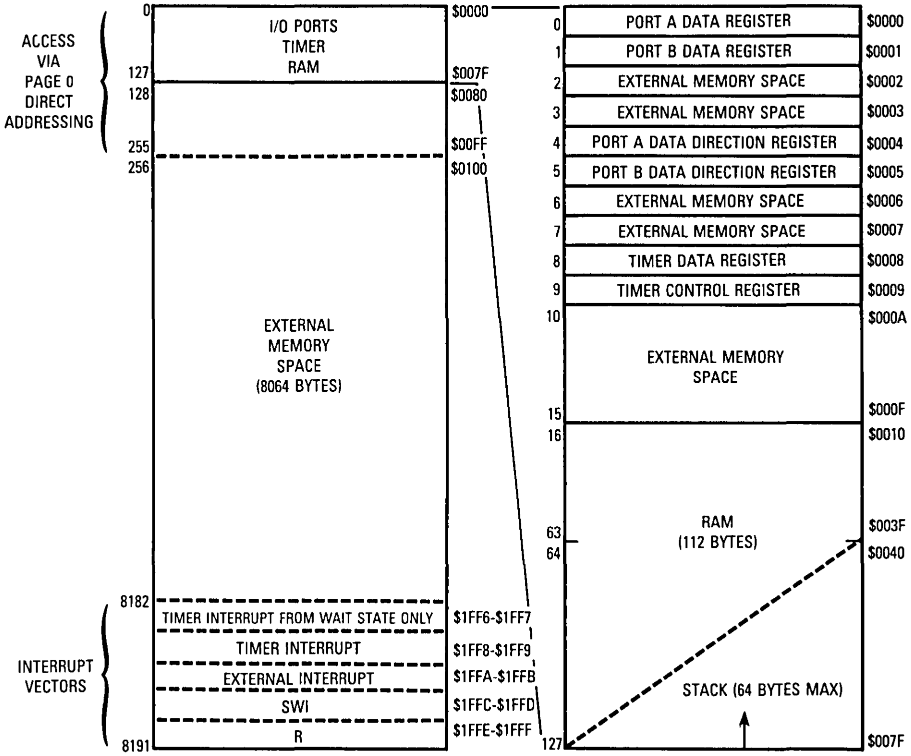

ACCESS VIA PAGE 0 DIRECT ADDRESSING
0
127
128
255
256
8182
8191
I/O PORTS
TIMER
RAM
EXTERNAL MEMORY SPACE (8064 BYTES)
TIMER INTERRUPT FROM WAIT STATE ONLY
TIMER INTERRUPT
EXTERNAL INTERRUPT
SWI
R
INTERRUPT VECTORS
$0000
$007F
$0080
$00FF
$0100
$1FF6-$1FF7
$1FF8-$1FF9
$1FFA-$1FFB
$1FFC-$1FFD
$1FFE-$1FFF
0 PORT A DATA REGISTER $0000
1 PORT B DATA REGISTER $0001
2 EXTERNAL MEMORY SPACE $0002
3 EXTERNAL MEMORY SPACE $0003
4 PORT A DATA DIRECTION REGISTER $0004
5 PORT B DATA DIRECTION REGISTER $0005
6 EXTERNAL MEMORY SPACE $0006
7 EXTERNAL MEMORY SPACE $0007
8 TIMER DATA REGISTER $0008
9 TIMER CONTROL REGISTER $0009
10 $000A
EXTERNAL MEMORY SPACE
15 $000F
16 $0010
RAM (112 BYTES)
63 $003F
64 $0040
STACK (64 BYTES MAX)
127 $007F

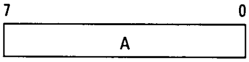
Figure 4. Memory Map

REGISTERS

The MPU contains the registers described in the following paragraphs.

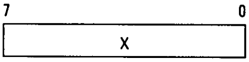
ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



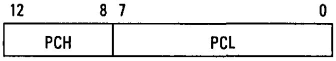
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

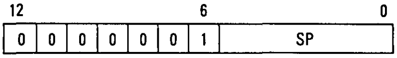
The program counter is an 13-bit register that contains the address of the next byte to be fetched.


12
8
7
0
PCH
PCL

STACK POINTER (SP)

The stack pointer is an 13-bit register that contains the address of the next free location on the stack. During an MPU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $007F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The seven most-significant bits of the stack pointer are permanently set at 0000001. Subroutines and interrupts may be nested down to location $0040 (64 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).


12
6
0
0
0
0
0
0
0
1
SP

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specifications can be taken as a result of their state. Each bit is explained in the following paragraphs.

4				0
H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic 1).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, and during shifts and rotates.

RESETS

The MPU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. There is a 1920 t_{cyc} after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 1920 t_{cyc}, the MPU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

EXTERNAL RESET INPUT

The MPU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

INTERRUPTS

The MPU can be interrupted three different ways: (1) through the external interrupt $\overline{\text{IRQ}}$ input pin, (2) with the internal timer interrupt request, or (3) using the software interrupt instruction (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack after which normal processing resumes. The stacking order is shown in Figure 5.

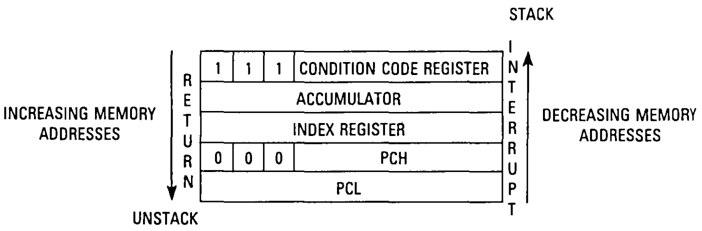
Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked, (I bit clear) proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction if the I bit is set (hardware interrupts masked). Refer to Figure 6 for the reset and interrupt instruction processing sequence.


STACK
INCREASING MEMORY ADDRESSES
RETURN
1 1 1 CONDITION CODE REGISTER
ACCUMULATOR
INDEX REGISTER
0 0 0 PCH
PCL
INTERRUPT
DECREASING MEMORY ADDRESSES
UNSTACK

NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 5. Interrupt Stacking Order

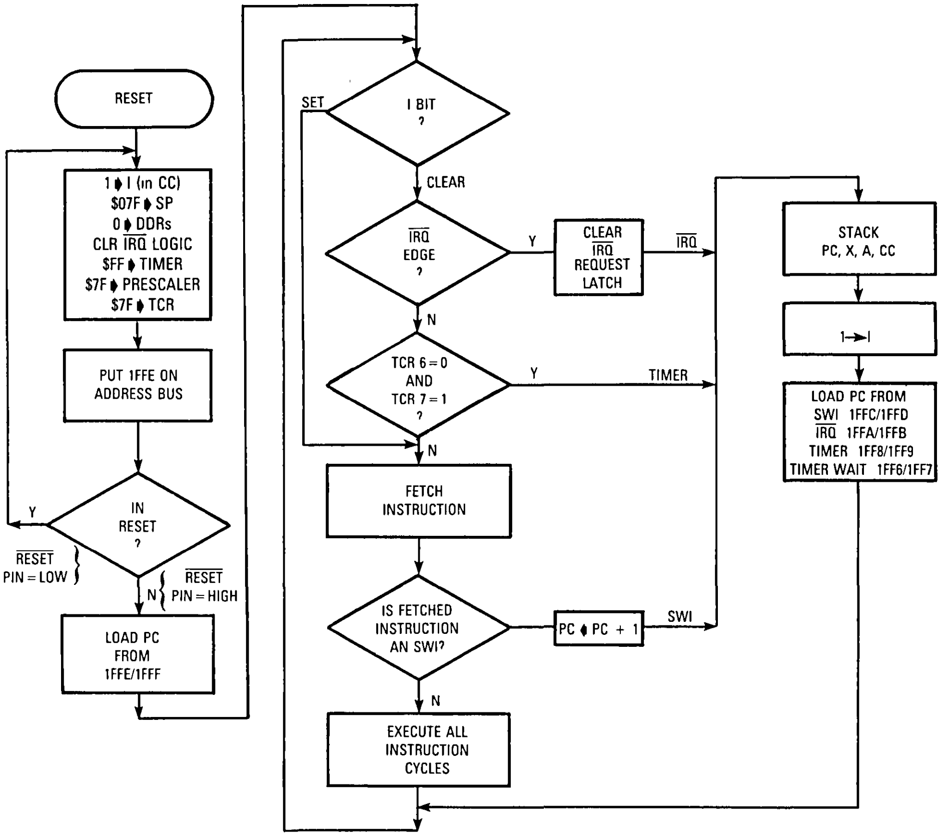

RESET
1 ➧ I (in CC)
$07F ➧ SP
0 ➧ DDRs
CLR IRQ LOGIC
$FF ➧ TIMER
$7F ➧ PRESCALER
$7F ➧ TCR
PUT 1FFE ON ADDRESS BUS
IN RESET ?
Y
RESET PIN = LOW
N
RESET PIN = HIGH
LOAD PC FROM 1FFE/1FFF
I BIT ?
SET
CLEAR
IRQ EDGE ?
Y
CLEAR IRQ REQUEST LATCH
IRQ
N
TCR 6 = 0 AND TCR 7 = 1 ?
Y
TIMER
N
FETCH INSTRUCTION
IS FETCHED INSTRUCTION AN SWI?
PC ➧ PC + 1
SWI
N
EXECUTE ALL INSTRUCTION CYCLES
STACK PC, X, A, CC
1 ➞ I
LOAD PC FROM
SWI 1FFC/1FFD
IRQ 1FFA/1FFB
TIMER 1FF8/1FF9
TIMER WAIT 1FF6/1FF7

Figure 6. Reset and Interrupt Processing Flowchart

TIMER INTERRUPT

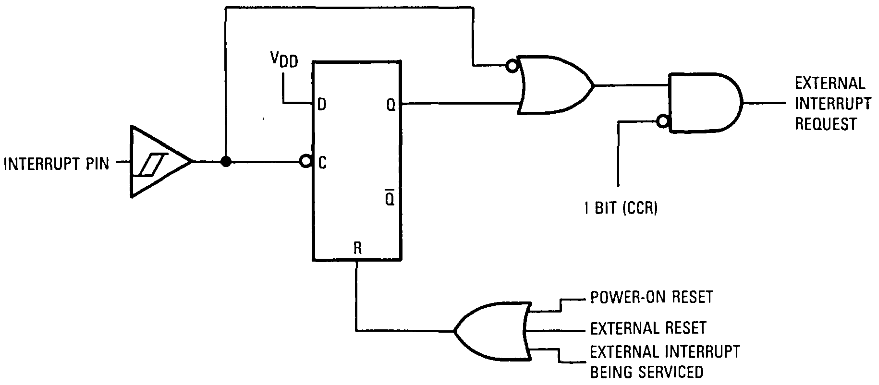
If the timer mask bit (TCR6) is cleared, each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack, and the I bit in the CCR is set masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

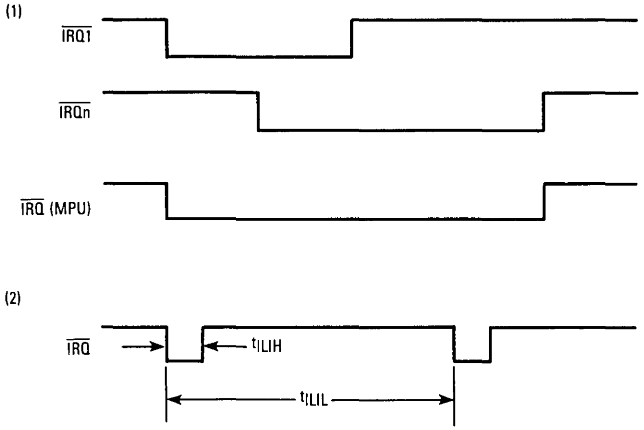
If the interrupt mask bit (I bit) of the condition code register is set, all interrupts are disabled. Clearing the I bit enables the external interrupts. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally, and the service routine address is specified by the contents of $1FFA and $1FFB.

Figure 7 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 20 instructions cycles to the total number of cycles it takes to complete the service routine. The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

(A) INTERRUPT FUNCTIONAL DIAGRAM



(B) INTERRUPT MODE DIAGRAM



Wire ORed Condition
If after servicing an interrupt the IRQ remains low, then the next interrupt is recognized.

Pulse Condition
The minimum pulse width (t_{ILIH} is one t_{cyc} The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles

Figure 7. External Interrupt

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI execution is similar to the hardware interrupts.

INTERRUPT CLARIFICATION

Under certain circumstances, the MPU (BP4XXXX and AW9XXXX) IRQ interrupt does not conform to the operation described in this documentation.

The level sensitive IRQ mode is fully operational, and most applications are unaffected. Under certain conditions the edge-triggered IRQ *might not be serviced*; therefore, it is recommended that the edge-triggered mode not be used.

An interrupt-vector address *can be* improperly generated in some circumstances. There is a possibility that when an external interrupt and timer interrupt occur during the WAIT mode, address locations $1FF2 and $1FF3 are selected instead of vector locations $1FF6 and $1FF7. If the WAIT mode is not used or the WAIT mode is used without external interrupt, no precautions are necessary. If the WAIT mode is used, the vector in locations $1FF6 and $1FF7 should be duplicated in $1FF2 and $1FF3. Therefore, normal program execution would not be disturbed if the wrong vector was selected.

LOW-POWER MODES

STOP

The STOP instruction places the MPU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 8.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt request and to disable any further time interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can only be brought out of the STOP mode by an external interrupt or reset.

WAIT

The WAIT instruction places the MPU in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 9. Thus, all internal processing is halted; however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt

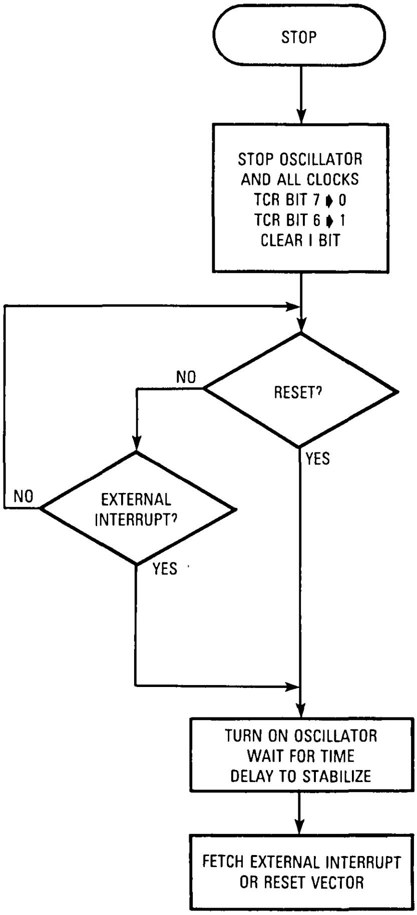

STOP
STOP OSCILLATOR
AND ALL CLOCKS
TCR BIT 7 ➧ 0
TCR BIT 6 ➧ 1
CLEAR I BIT
RESET?
NO
YES
EXTERNAL
INTERRUPT?
NO
YES
TURN ON OSCILLATOR
WAIT FOR TIME
DELAY TO STABILIZE
FETCH EXTERNAL INTERRUPT
OR RESET VECTOR

Figure 8. STOP Function Flowchart

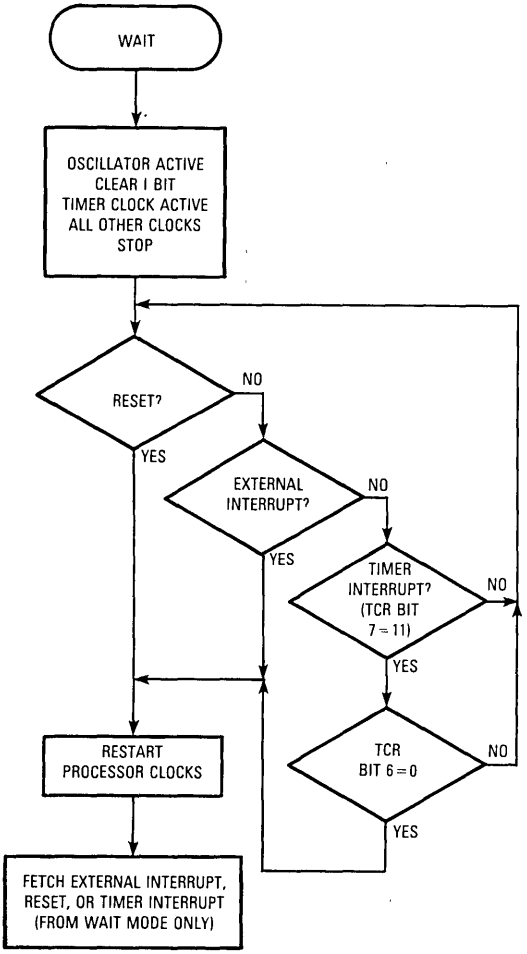

WAIT
OSCILLATOR ACTIVE
CLEAR I BIT
TIMER CLOCK ACTIVE
ALL OTHER CLOCKS
STOP
RESET?
NO
YES
EXTERNAL
INTERRUPT?
NO
YES
TIMER
INTERRUPT?
(TCR BIT
7=1)
NO
YES
TCR
BIT 6=0
NO
YES
RESTART
PROCESSOR CLOCKS
FETCH EXTERNAL INTERRUPT,
RESET, OR TIMER INTERRUPT
(FROM WAIT MODE ONLY)

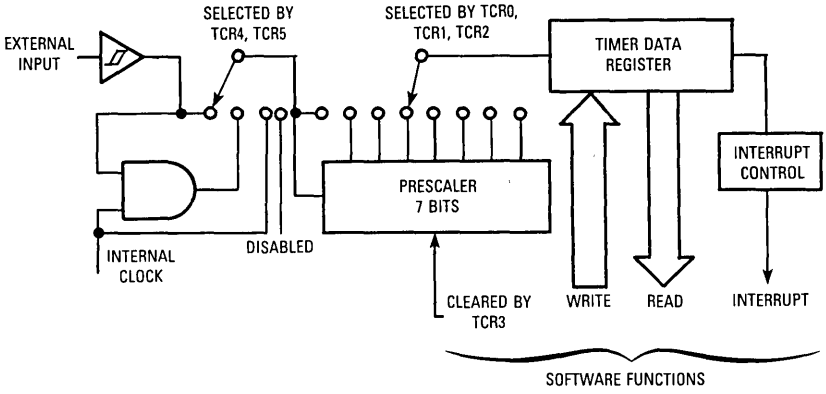
Figure 9. WAIT Function Flowchart

(not the timer wait interrupt) is serviced since the MPU is no longer in the WAIT mode.

TIMER

The MPU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The various timer sources are made via the timer control register (TCR). The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 10 for timer block diagram.

The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared, the processor receives the interrupt. The MPU responds to this interrupt by 1) saving the present CPU state on the stack, 2) fetching the timer interrupt vector, and 3) executing the interrupt routine. The timer interrupt request bit must be cleared by software. Refer to **RESETS** and **INTERRUPTS** for additional information.



NOTES
1 Prescaler and timer data register are clocked on the falling edge of the internal clock (AS) or external input
2 Timer data register is written to during data strobe (DS) and counts down continuously

Figure 10. Timer Block Diagram

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic 1; however, the TCR bit 3 always reads as a logic 0 to ensure proper operation with read-modify-write instructions.

The timer continues to count past zero, falling from $00 through $FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process.

SOFTWARE CONTROLLED MODE

The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TCR4 and TCR5). The following paragraphs describe the different modes.

Timer Input Mode 1

When TCR4 and TCR5 are both programmed to zero, the timer input is from the internal clock (phase 2) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement. During the WAIT instruction, the internal clock to the timer continues to run at its normal rate.

Timer Input Mode 2

When TCR4 = 1 and TCR5 = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is ±1.

Timer Input Mode 3

When TCR4 = 0 and TCR5 = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

Timer Input Mode 4

When TCR4 and TCR5 are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts.

TIMER CONTROL REGISTER (TCR) $009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3.

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

RESET:

0	1	U	U	U	U	U	U

TCR7 — Timer Interrupt Request

Used to indicate the timer interrupt when it is logic one.

1 = Set when the timer data register changes to all zeros.

0 = Cleared by external reset, power-on reset, or under program control.

TCR6 — Timer Interrupt Mask

Used to inhibit the timer interrupt.

1 = Interrupt inhibited.

0 = Interrupt enabled.

TCR5 — External or Internal
Selects input clock source.
1 = External clock selected.
0 = Internal clock selected ($f_{OSC}/4$).

TCR4 — TIMER External Enable
Used to enable external TIMER pin or to enable the internal clock.
1 = Enables external timer pin.
0 = Disables external timer pin.

TCR3 — Prescaler Clear
Write only bit. Writing a 1 to this bit resets the prescaler to zero. A read of this location always indicates a zero.

TCR2, TCR1, TCR0 — Prescaler select bits
Decoded to select one of eight outputs of the prescaler.

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MPU has a set of 61 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction listing.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

— Continued —

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following listing of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

BIT MANIPULATION INSTRUCTIONS

The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following listing of bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n = 0 . . . 7)
Set Bit n	BSET n (n = 0 . . . 7)
Clear Bit n	BCLR n (n = 0 . . . 7)

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two byte instructions. Refer to the following listing of branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following listing of control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 2 is an opcode map for the instructions used on the MPU.

ADDRESSING MODES

The MPU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code coversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

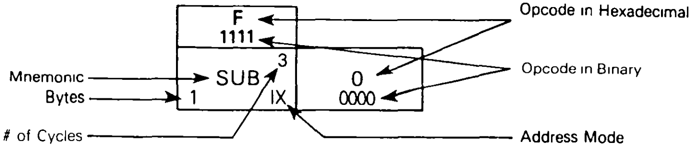
Table 2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL								2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



INDEX, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this 2-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the direct addressing of the byte to which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports, A and B are write only registers (registers at $0004 and $0005). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, these instructions cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

BIT TEST and BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805E2 MC146805E2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation it is recommended the V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		100	
Quad Package		100	

DC ELECTRICAL CHARACTERISTICS @ 3.0 V (V_{DD}=3.0 Vdc, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage (I_{Load}≤10.0 μA)	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Total Supply Current (C_L = 50 pF – No dc Loads, t_{cyc} = 5 μs) Run (V_{IL} = 0.2 V, V_{IH} = V_{DD}−0.2 V) Wait (Test Conditions — See Note Below) Stop (Test Conditions — See Note Below)	 I_{DD} I_{DD} I_{DD}	 — — —	 1.3 200 100	 mA μA μA
Output High Voltage (I_{Load}=0 25 mA) A8-A12, B0-B7, DS, AS, R/$\overline{\text{W}}$ (I_{Load} = 0.1 mA) PA0-PA7, PB0-PB7	 V_{OH} V_{OH}	 2.7 2.7	 — —	 V V
Output Low Voltage (I_{Load}=0.25 mA) A8-A12, B0-B7, PB0-PB7, DS, AS, R/$\overline{\text{W}}$, PA0-PA7	V_{OL}	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, B0-B7 TIMER, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$ OSC1	 V_{IH} V_{IH} V_{IH}	 2.1 2.5 2.1	 — — —	 V V V
Input Low Voltage (All Inputs)	V_{IL}	—	0.5	V
Frequency of Operation Crystal External Clock	 f_{osc} f_{osc}	 — dc	 1.0 1.0	 MHz MHz
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	I_{in}	—	±1	μA
Hi-Z Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	—	±10	μA
Capacitance $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER	C_{in}	—	8.0	pF
Capacitance DS, AS, R/$\overline{\text{W}}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	—	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL}=0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH}=V_{DD}−0.2 V for $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, and TIMER
OSC1 input is a squarewave from V_{SS}+0.2 V to V_{DD} − 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode I_{DD} is affected linearly by this capacitance.

DC ELECTRICAL CHARACTERISTICS @ 5.0 V (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10.0\ \mu A$)	V_{OL} V_{OH}	— $V_{DD} - 0.1$	0.1 —	V
Total Supply Current (C_L = 130 pF — On Bus, C_L = 50 pF — On Ports, No dc Loads, t_{cyc} = 1.0 µs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V)				
Run	I_{DD}	—	10	mA
Wait (Test Conditions — See Note Below)	I_{DD}	—	1.5	mA
Stop (Test Conditions — See Note Below)	I_{DD}	—	200	µA
Output High Voltage				
(I_{Load} = 1.6mA) A8-A12, B0-B7, DS, AS, R/$\overline{W}$	V_{OH}	4.1	—	V
(I_{Load} = 0.36 mA) PA0-PA7, PB0-PB7	V_{OH}	4.1	—	V
Output Low Voltage (I_{Load} = 1.6 mA) A8-A12, B0-B7, PA0-PA7, PB0-PB7, DS, AS, R/$\overline{W}$	V_{OL}	—	0.4	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	$V_{DD} - 2.0$	—	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$V_{DD} - 0.8$	—	V
OSC1	V_{IH}	$V_{DD} - 1.5$	—	V
Input Low Voltage (All Inputs)	V_{IL}	—	0.8	V
Frequency of Operation				
Crystal	f_{osc}	—	5.0	MHz
External Clock	f_{osc}	dc	5.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	—	±1	µA
Hi-Z Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	—	±10	µA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, TIMER	C_{in}	—	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	—	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
$V_{IH} = V_{DD} - 0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER.
OSC1 input is a squarewave from $V_{SS} + 0.2$ V to $V_{DD} - 0.2$ V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode (I_{DD}) is affected linearly by this capacitance.

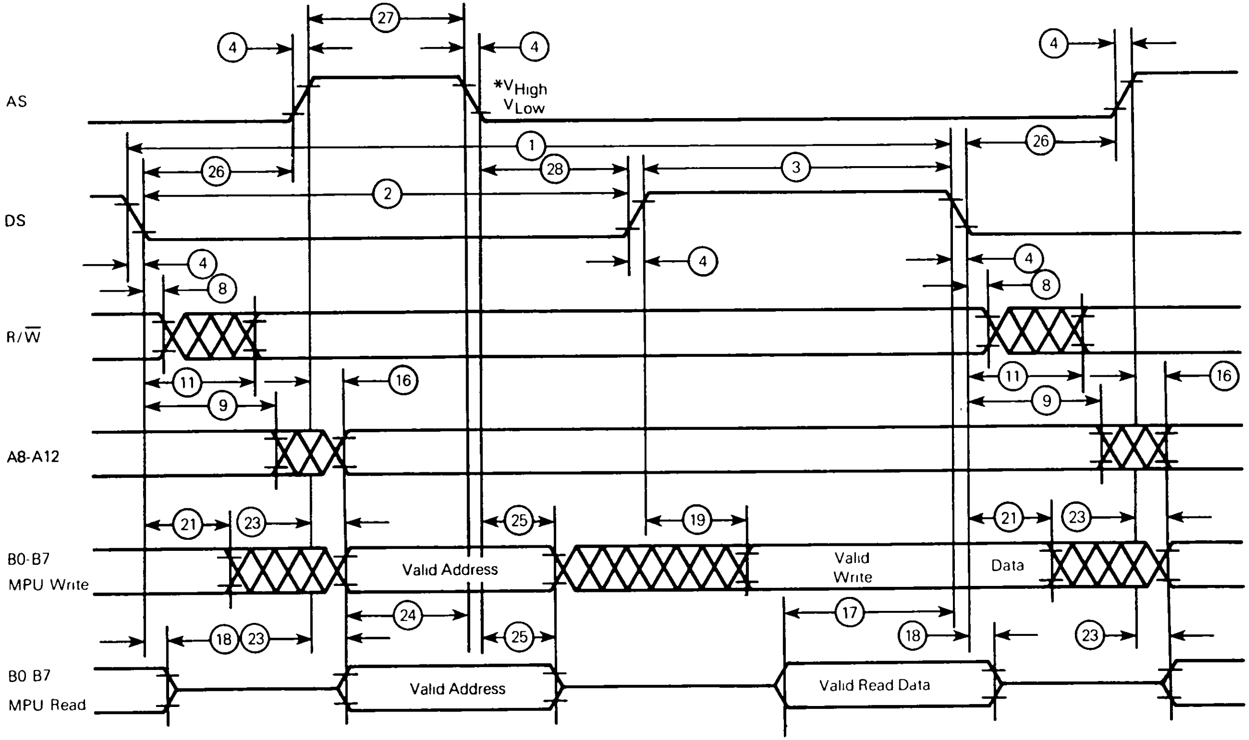
Table 3. Control Timing ($V_{SS}=0$, $T_A=T_L$ to T_H)

Characteristic	Symbol	$V_{DD}=3.0$ V $f_{osc}=1$ MHz			$V_{DD}=5.0$ V ± 10% $f_{osc}=5.0$ MHz			Unit
		Min	Typ	Max	Min	Typ	Max	
I/O Port Timing — Input Setup Time (Figure 16)	t_{PVASL}	500	—	—	250	—	—	ns
Input Hold Time (Figure 16)	t_{ASLPX}	100	—	—	100	—	—	ns
Output Delay Time (Figure 16)	t_{ASLPV}	—	—	0	—	—	0	ns
Interrupt Setup Time (Figure 13)	t_{ILASL}	2	—	—	0.4	—	—	µs
Crystal Oscillator Startup Time (Figure 12)	t_{OXOV}	—	30	300	—	15	100	ms
Wait Recovery Startup Time (Figure 14)	t_{IVASH}	—	—	10	—	—	2	µs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 15)	t_{ILASH}	—	30	300	—	15	100	ms
Required Interrupt Release (Figure 13)	t_{DSLIH}	—	—	5	—	—	1.0	µs
Timer Pulse Width (Figure 14)	t_{TH}, t_{TL}	0.5	—	—	0.5	—	—	t_{cyc}
Reset Pulse Width (Figure 12)	t_{RL}	5.5	—	—	1.5	—	—	µs
Timer Period (Figure 14)	t_{TLTL}	1.0	—	—	1.0	—	—	t_{cyc}
Interrupt Pulse Width Low (Figure 7)	t_{ILIH}	1.0	—	—	1.0	—	—	t_{cyc}
Interrupt Pulse Period (Figure 7)	t_{ILIL}	*	—	—	*	—	—	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	—	—	200	—	—	ns
OSC1 Pulse Width High	t_{OH}	350	—	—	75	—	—	ns
OSC1 Pulse Width Low	t_{OL}	350	—	—	75	—	—	ns

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Table 4. Bus Timing ($T_A=T_L$ to T_H, $V_{SS}=0$ V) See Figure 11.

Num	Characteristics	$f_{osc}=1$ MHz $V_{DD}=3.0$ V 50 pF Load		$f_{osc}=5$ MHz $V_{DD}=5.0$ V ±10%, 1 TTL and 130 pF Load		Unit
		Min	Max	Min	Max	
1	Cycle Time	5000	dc	1000	dc	ns
2	Pulse Width, DS Low	2800	—	560	—	ns
3	Pulse Width, DS High	1800	—	375	—	ns
4	Clock Transition	—	100	—	30	ns
8	R/$\overline{W}$	10	—	10	—	ns
9	Non-Muxed Address Hold	800	—	100	—	ns
11	R/$\overline{W}$ Delay from DS Fall	—	500	—	300	ns
16	Non-Muxed Address Delay from AS Rise	0	200	0	100	ns
17	MPU Read Data Setup	200	—	115	—	ns
18	Read Data Hold	0	800	0	160	ns
19	MPU Data Delay, Write	—	0	—	120	ns
21	Write Data Hold	800	—	55	—	ns
23	Muxed Address Delay from AS Rise	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	600	—	55	—	ns
25	Muxed Address Hold	250	750	60	180	ns
26	Delay DS Fall to AS Rise	800	—	160	—	ns
27	Pulse Width, AS High	850	—	175	—	ns
28	Delay, AS Fall to DS Rise	800	—	160	—	ns


AS
DS
R/$\overline{W}$
A8-A12
B0-B7
MPU Write
B0 B7
MPU Read
*VHigh
VLow
Valid Address
Valid Address
Valid Write
Data
Valid Read Data
①
②
③
④
⑧
⑨
⑪
⑯
⑰
⑱
⑲
㉑
㉓
㉔
㉕
㉖
㉗
㉘

NOTES:
*V_{High} = 2.0 V, V_{Low} = 0.5 V for V_{DD} = 3 V for outputs only.
V_{High} = V_{DD} − 2.0 V, V_{Low} = 0.8 V for V_{DD} = 5 V ± 10% for outputs only.

Refer to Table 4 for Number Reference.

Figure 11. MC146805E2 Bus Timing

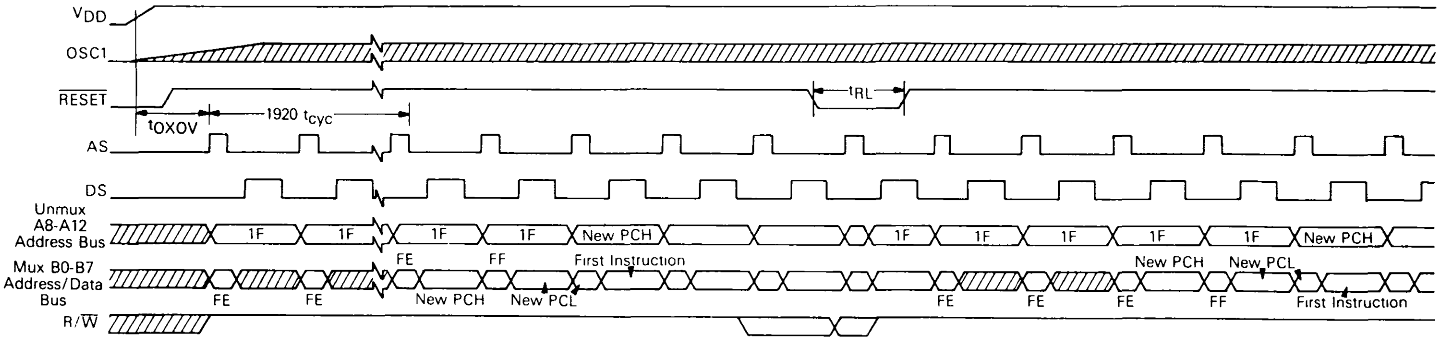
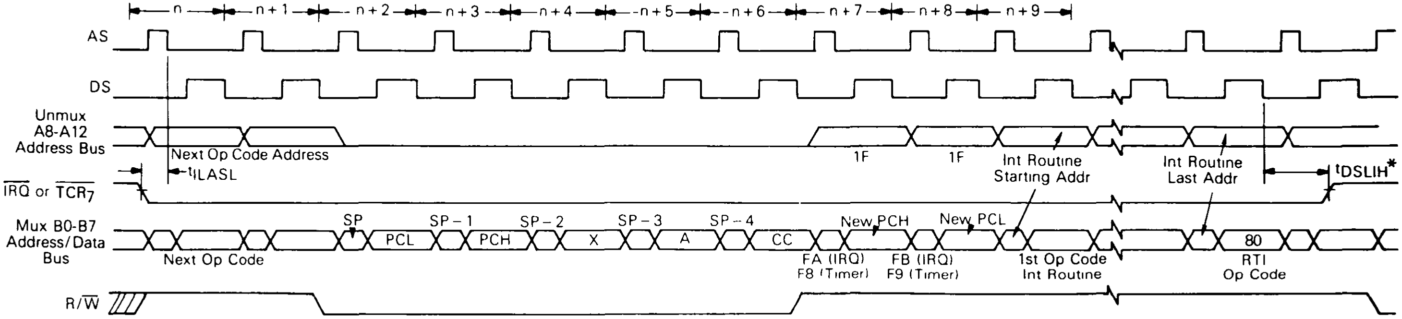

VDD
OSC1
RESET
AS
DS
Unmux A8-A12 Address Bus
Mux B0-B7 Address/Data Bus
R/W
tOXOV
1920 tcyc
tRL
1F
New PCH
FE
FF
First Instruction
New PCL

Figure 12. Power-on Reset and $\overline{\text{RESET}}$ Timing


n
n + 1
n + 2
n + 3
n + 4
n + 5
n + 6
n + 7
n + 8
n + 9
AS
DS
Unmux A8-A12 Address Bus
Next Op Code Address
tILASL
1F
Int Routine Starting Addr
Int Routine Last Addr
tDSLIH*
IRQ or TCR7
Mux B0-B7 Address/Data Bus
SP
SP – 1
SP – 2
SP – 3
SP – 4
PCL
PCH
X
A
CC
New PCH
New PCL
80
Next Op Code
FA (IRQ)
F8 (Timer)
FB (IRQ)
F9 (Timer)
1st Op Code Int Routine
RTI Op Code
R/W

*t_{DSLIH} – The interrupting device must release the $\overline{\text{IRQ}}$ line within this time to prevent subsequent recognition of the same interrupt

Figure 13. $\overline{\text{IRQ}}$ and $\overline{\text{TCR7}}$ Interrupt Timing

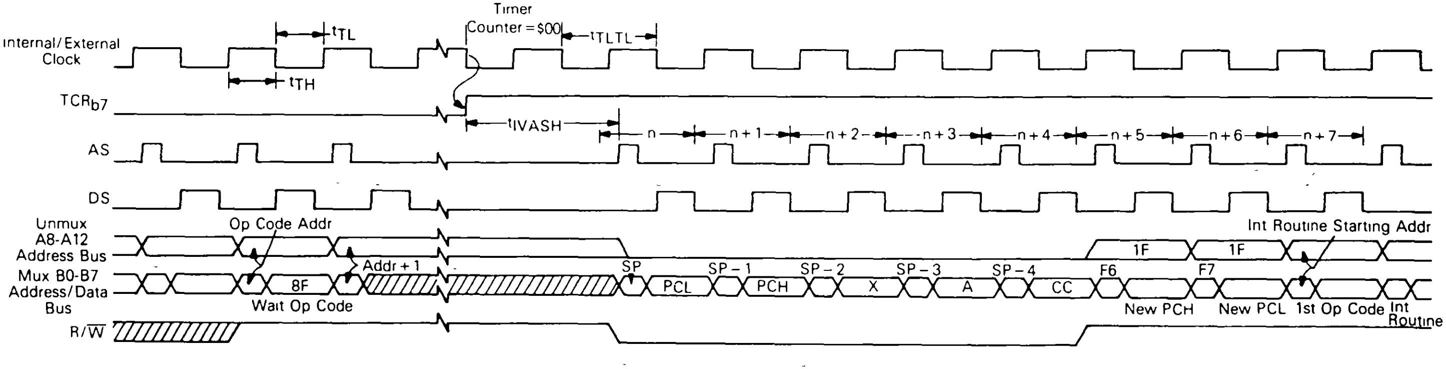
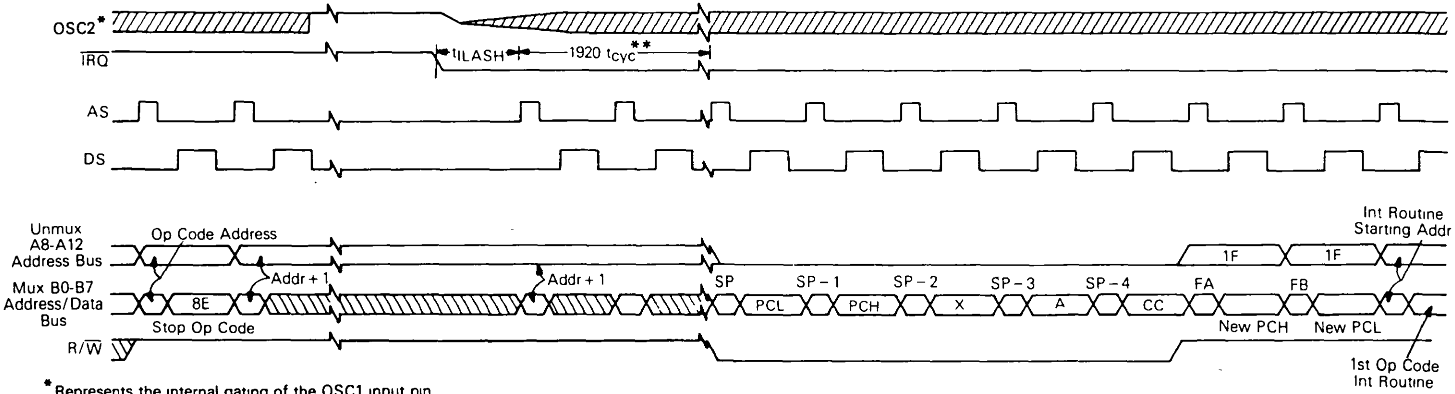

Internal/External Clock
tTL
tTH
Timer Counter = $00
tTLTL
TCRb7
tIVASH
n
n + 1
n + 2
n + 3
n + 4
n + 5
n + 6
n + 7
AS
DS
Unmux A8-A12 Address Bus
Op Code Addr
Int Routine Starting Addr
1F
1F
Mux B0-B7 Address/Data Bus
Addr + 1
8F
Wait Op Code
SP
PCL
SP − 1
PCH
SP − 2
X
SP − 3
A
SP − 4
CC
F6
New PCH
F7
New PCL
1st Op Code
Int Routine
R/W

Figure 14. Timer Interrupt After WAIT Instruction: Timing

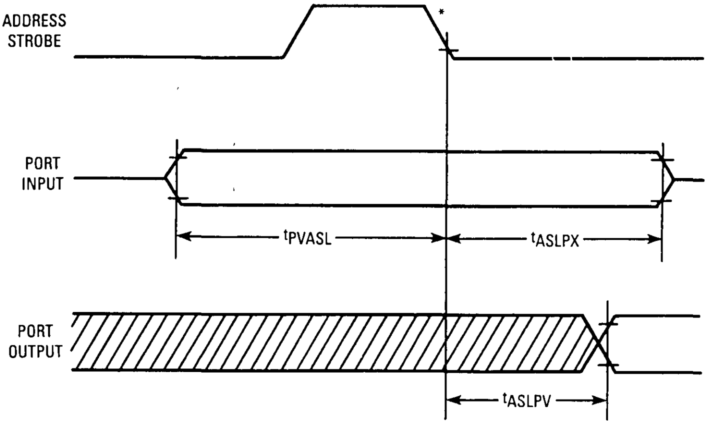

OSC2*
IRQ
tILASH
1920 tcyc**
AS
DS
Unmux A8-A12 Address Bus
Op Code Address
1F
1F
Int Routine Starting Addr
Mux B0-B7 Address/Data Bus
8E
Addr + 1
Addr + 1
Stop Op Code
SP
PCL
SP − 1
PCH
SP − 2
X
SP − 3
A
SP − 4
CC
FA
New PCH
FB
New PCL
R/W
1st Op Code Int Routine

*Represents the internal gating of the OSC1 input pin

** t_{cyc} is one instruction cycle (for $f_{osc} = 5$ MHz, $t_{cyc} = 1$ µs)

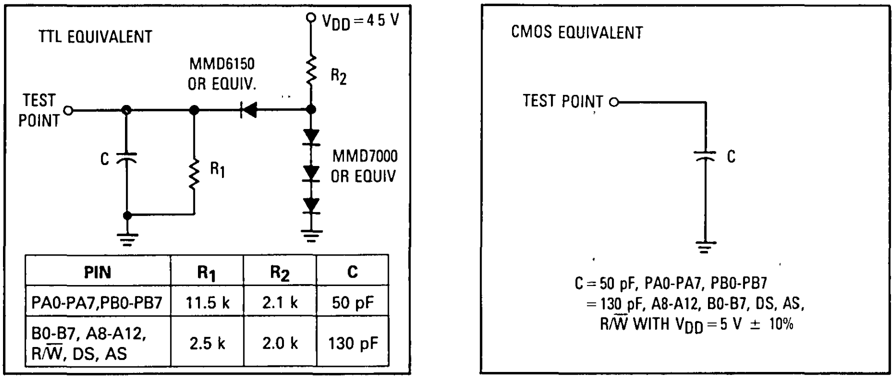
Figure 15. Interrupt Recovery From STOP Instruction: Timing

$V_{Low}=0.8$ V, $V_{High}=V_{DD}-2.0$ V, $V_{DD}=5.0 \pm 10\%$
$T_A=T_L$ to T_H, C_L on Port$=50$ pF, $f_{osc}=5$ MHz)


ADDRESS STROBE
*
PORT INPUT
t_{PVASL}
t_{ASLPX}
PORT OUTPUT
t_{ASLPV}

*The address strobe of the first cycle of the next instruction.

Figure 16. I/O Port Timing


TTL EQUIVALENT
$V_{DD}=4.5$ V
MMD6150
OR EQUIV.
R_2
TEST
POINT
C
R_1
MMD7000
OR EQUIV
CMOS EQUIVALENT
TEST POINT
C
C = 50 pF, PA0-PA7, PB0-PB7
= 130 pF, A8-A12, B0-B7, DS, AS,
R/$\overline{W}$ WITH $V_{DD}=5$ V ± 10%

PIN	R_1	R_2	C
PA0-PA7,PB0-PB7	11.5 k	2.1 k	50 pF
B0-B7, A8-A12, R/$\overline{W}$, DS, AS	2.5 k	2.0 k	130 pF

Figure 17. Equivalent Test Loads

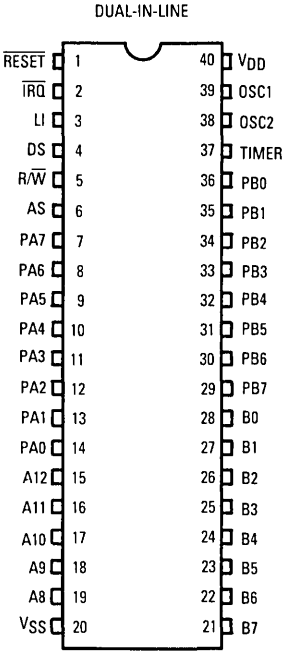
ORDERING INFORMATION

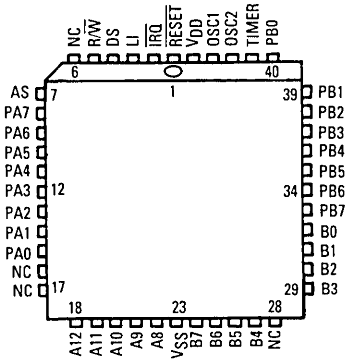
The following table provides generic information pertaining to the package type, temperature, and MC part numbers for the MC146805E2.

Table 5. Generic Information

Package Type	Frequency (MHz)	Temperature	Part Number
Plastic P Suffix	1.0 1.0	0°C to 70°C −40°C to +85°C	MC146805E2P MC146805E2CP
PLCC FN Suffix	1.0 1.0	0°C to 70°C −40°C to +85°C	MC146805E2FN MC146805E2CFN

PIN ASSIGNMENT


DUAL-IN-LINE
RESET 1
IRQ 2
LI 3
DS 4
R/W 5
AS 6
PA7 7
PA6 8
PA5 9
PA4 10
PA3 11
PA2 12
PA1 13
PA0 14
A12 15
A11 16
A10 17
A9 18
A8 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TIMER
36 PB0
35 PB1
34 PB2
33 PB3
32 PB4
31 PB5
30 PB6
29 PB7
28 B0
27 B1
26 B2
25 B3
24 B4
23 B5
22 B6
21 B7


NC
R/W
DS
LI
IRQ
RESET
VDD
OSC1
OSC2
TIMER
PB0
6
40
1
AS 7
PA7
PA6
PA5
PA4
PA3 12
PA2
PA1
PA0
NC
NC 17
39 PB1
PB2
PB3
PB4
PB5
34 PB6
PB7
B0
B1
B2
29 B3
18
23
28
A12
A11
A10
A9
A8
VSS
B7
B6
B5
B4
NC

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146805F2

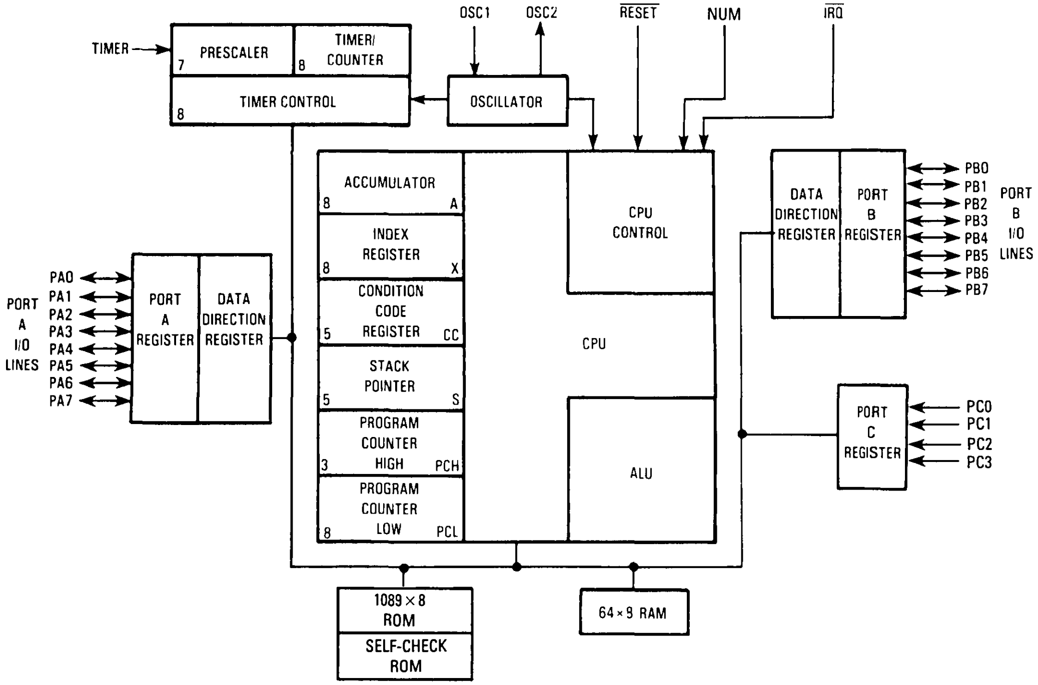
Technical Summary
8-Bit Microcomputer Unit

The MC146805F2 (CMOS) Microcomputer Unit (MCU) is a member of the MC146805 Family of microcomputers. This low cost and low power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual* (M6805UM(AD2)) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Internal 8-Bit Timer with 7-Bit Programmable Prescaler
- On-chip Clock
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Self-check
- Power-saving STOP and WAIT Modes
- Single 3.0- to 6.0-Volt Supply
- Fully Static Operation
- 1089 Bytes of ROM
- 64 Bytes of RAM
- 20 I/O Ports

BLOCK DIAGRAM


OSC1
OSC2
$\overline{\text{RESET}}$
NUM
$\overline{\text{IRQ}}$
TIMER
PRESCALER
7
TIMER/ COUNTER
8
TIMER CONTROL
8
OSCILLATOR
ACCUMULATOR
8
A
INDEX REGISTER
8
X
CONDITION CODE REGISTER
5
CC
STACK POINTER
5
S
PROGRAM COUNTER HIGH
3
PCH
PROGRAM COUNTER LOW
8
PCL
CPU CONTROL
CPU
ALU
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A I/O LINES
PORT A REGISTER
DATA DIRECTION REGISTER
DATA DIRECTION REGISTER
PORT B REGISTER
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PORT B I/O LINES
PORT C REGISTER
PC0
PC1
PC2
PC3
1089 × 8 ROM
SELF-CHECK ROM
64 × 8 RAM

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcomputer using these two pins. V_{DD} is +5 volts (±0.5Δ) power, and V_{SS} is ground.

NUM

This pin is intended for use in self-check only. In normal operation, this pin is connected to V_{SS}. A resistor of up to 10 kilohms to ground may be used if the oscillator is 32.768 KHz or less.

$\overline{\text{IRQ}}$

This pin is a photomask option with two different choices of interrupt triggering sensitivity; level and negative edge or negative edge only. Refer to **INTERRUPTS** for more detailed information.

OSC1, OSC2

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal is connected to these pins to provide a system clock.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(b). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(a) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

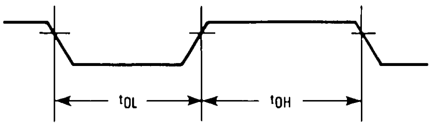
External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(c). This option may only be used with the crystal oscillator option. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

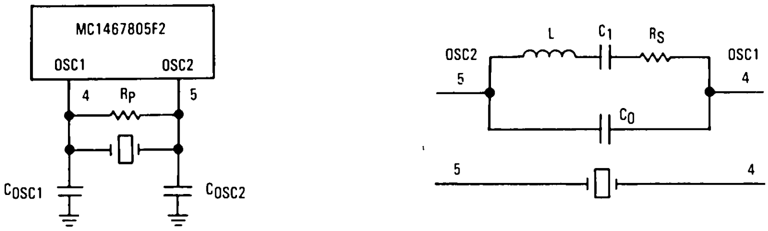
CRYSTAL PARAMETERS

	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

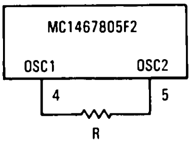
(a) OSCILLATOR WAVEFORM



(b) CRYSTAL OSCILLATOR CONNECTIONS AND EQUIVALENT CRYSTAL CIRCUIT



(c) RC OSCILLATOR CONNECTION*



*Approximately 10% to 25% accuracy (excludes resistor tolerance) external register

(d) EXTERNAL CLOCK SOURCE CONNECTIONS

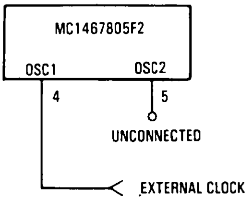


Figure 1. Oscillator Connections

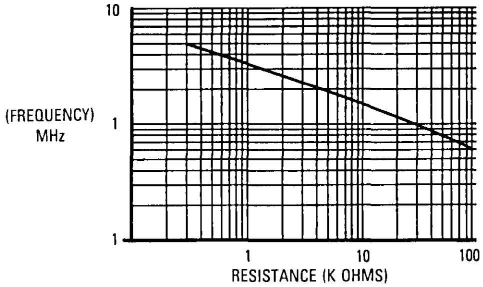


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

TIMER

This pin is used as an external input to control the internal timer/counter circuitry.

$\overline{\text{RESET}}$

This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). Port A and Port B are programmable as either inputs or outputs under software control of the data direction registers. Port C is fixed input ports and not controlled by any data register. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Ports A and B are programmable as either input or output under software control of the corresponding write-only data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output and a logic 0 for input. On reset, all the DDRs are initialized to a logic 0 state to put the ports in the input mode. The port output registers are not initialized on reset and should be written to before setting the DDR bits.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and also to the latched output when the DDR is an output (1). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

MEMORY

The MCU is capable of addressing 2048 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consist of self-check ROM, user ROM, user RAM, a timer control register, and I/O. The interrupt vectors are located from $7F8 to $7FF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the shared stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

7	0
A	

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{\text{W}}$ is an internal signal.

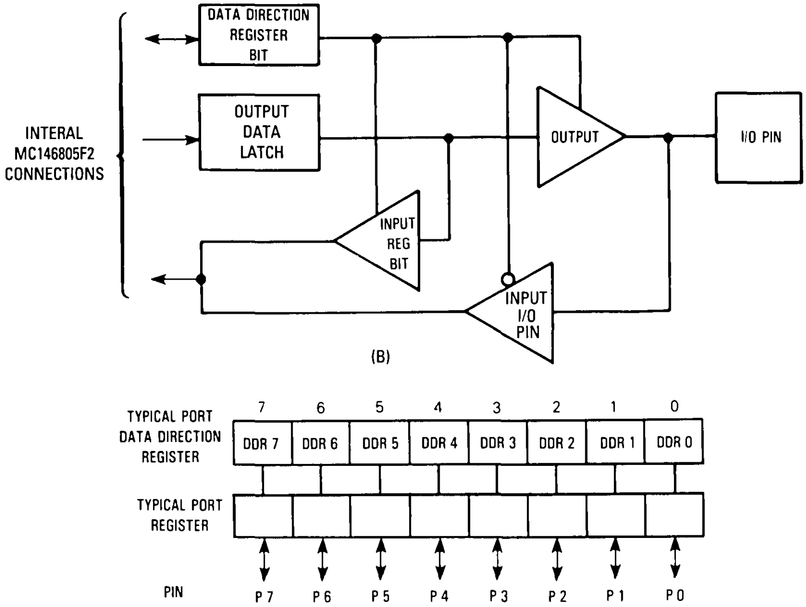

INTERAL
MC146805F2
CONNECTIONS
DATA DIRECTION REGISTER BIT
OUTPUT DATA LATCH
OUTPUT
I/O PIN
INPUT REG BIT
INPUT I/O PIN
(B)
TYPICAL PORT DATA DIRECTION REGISTER
7
6
5
4
3
2
1
0
DDR 7
DDR 6
DDR 5
DDR 4
DDR 3
DDR 2
DDR 1
DDR 0
TYPICAL PORT REGISTER
PIN
P7
P6
P5
P4
P3
P2
P1
P0

Figure 3. Typical Port I/O Circuitry and Register Configuration

may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.

7	0
X	

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next byte to be fetched.

10 8	7 0
PCH	PCL

STACK POINTER (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The six most-significant bits of the stack pointer are permanently set at 000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

10					5	4 0
0	0	0	0	1	1	SP

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specifications can be taken as a result of their state. Each bit is explained in the following paragraphs.

4				0
H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

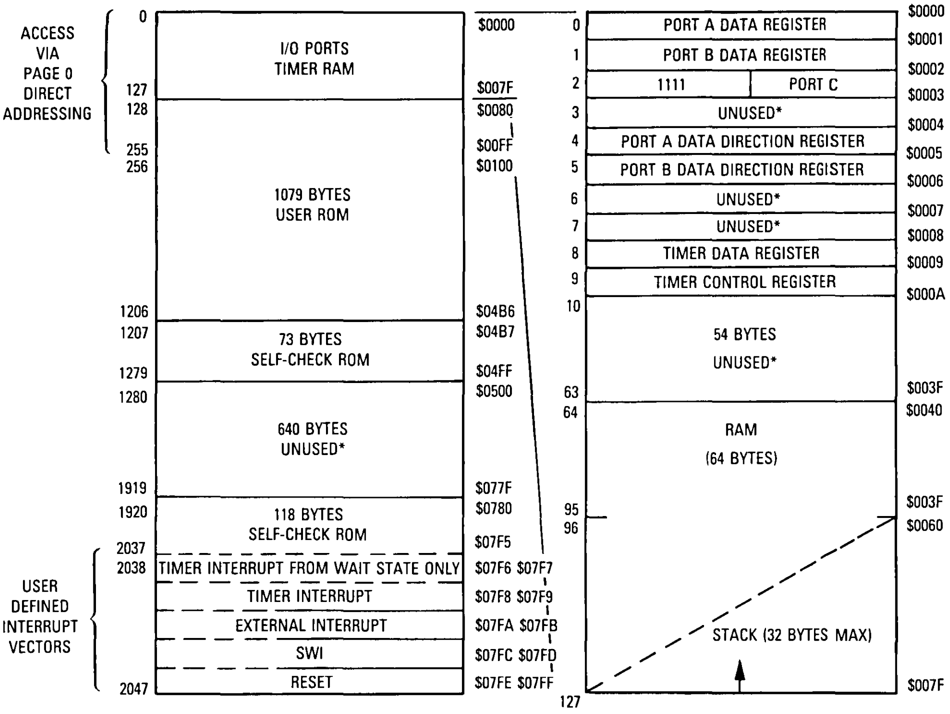
When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic 1).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.


ACCESS VIA PAGE 0 DIRECT ADDRESSING
0
127
128
255
256
1206
1207
1279
1280
1919
1920
2037
2038
2047
I/O PORTS
TIMER RAM
1079 BYTES
USER ROM
73 BYTES
SELF-CHECK ROM
640 BYTES
UNUSED*
118 BYTES
SELF-CHECK ROM
TIMER INTERRUPT FROM WAIT STATE ONLY
TIMER INTERRUPT
EXTERNAL INTERRUPT
SWI
RESET
USER DEFINED INTERRUPT VECTORS
$0000
$007F
$0080
$00FF
$0100
$04B6
$04B7
$04FF
$0500
$077F
$0780
$07F5
$07F6 $07F7
$07F8 $07F9
$07FA $07FB
$07FC $07FD
$07FE $07FF
0
1
2
3
4
5
6
7
8
9
10
63
64
95
96
127
PORT A DATA REGISTER
PORT B DATA REGISTER
1111
PORT C
UNUSED*
PORT A DATA DIRECTION REGISTER
PORT B DATA DIRECTION REGISTER
UNUSED*
UNUSED*
TIMER DATA REGISTER
TIMER CONTROL REGISTER
54 BYTES
UNUSED*
RAM
(64 BYTES)
STACK (32 BYTES MAX)
$0000
$0001
$0002
$0003
$0004
$0005
$0006
$0007
$0008
$0009
$000A
$003F
$0040
$003F
$0060
$007F

*READS OF UNUSED LOCATIONS UNDEFINED

Figure 4. Memory Map

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, and during shifts and rotates.

SELF CHECK

The self check is initiated by tying NUM and TIMER pins to a logic "1", and then executing a reset. The following tests are executed automatically:

I/O — Functionally exercise ports A, B, and C
RAM — Walking bit test
ROM — Exclusive OR with ODD "1s" parity result
Timer — Functionally exercise timer
Interrupts — Functionally exercise external and timer interrupts

The RAM self check, ROM checksum, and timer test are available to the user and do not require any external hardware.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input (RESET). The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. There is a 1920 t_{cyc} delay after the oscillator becomes active. If the RESET pin is low at the end of 1920 t_{cyc}, the MCU will remain in the reset condition until RESET goes high.

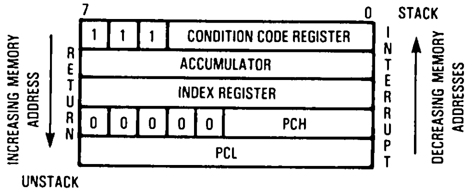
EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

INTERRUPTS

The MCU can be interrupted three different ways: (1) through the external interrupt IRQ input pin, (2) with the internal timer interrupt request, or (3) using the software interrupt instruction (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack, and then normal processing resumes. The stacking order is shown in Figure 5.



NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 5. Interrupt Stacking Order

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked (I bit clear) proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction if the I bit is set (hardware interrupts masked). Refer to Figure 6 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the CCR is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the condition code register is set, all interrupts are disabled. Clearing the I bit enables the external interrupts. The external interrupt is internally synchronized and then latched on the falling edge of IRQ. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at IRQ is latched internally, and the service routine address is specified by the contents of $7FA and $7FB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as mask options. Figure 7 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 20 instructions cycles to the total number of cycles it takes to complete the service routine. The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI is executed similar to the hardware interrupts.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 8.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt request and to disable any further time interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can only be brought out of the STOP mode by an external interrupt or reset.

WAIT

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 9. Thus, all internal processing is halted; however, the timer continues to count normally.

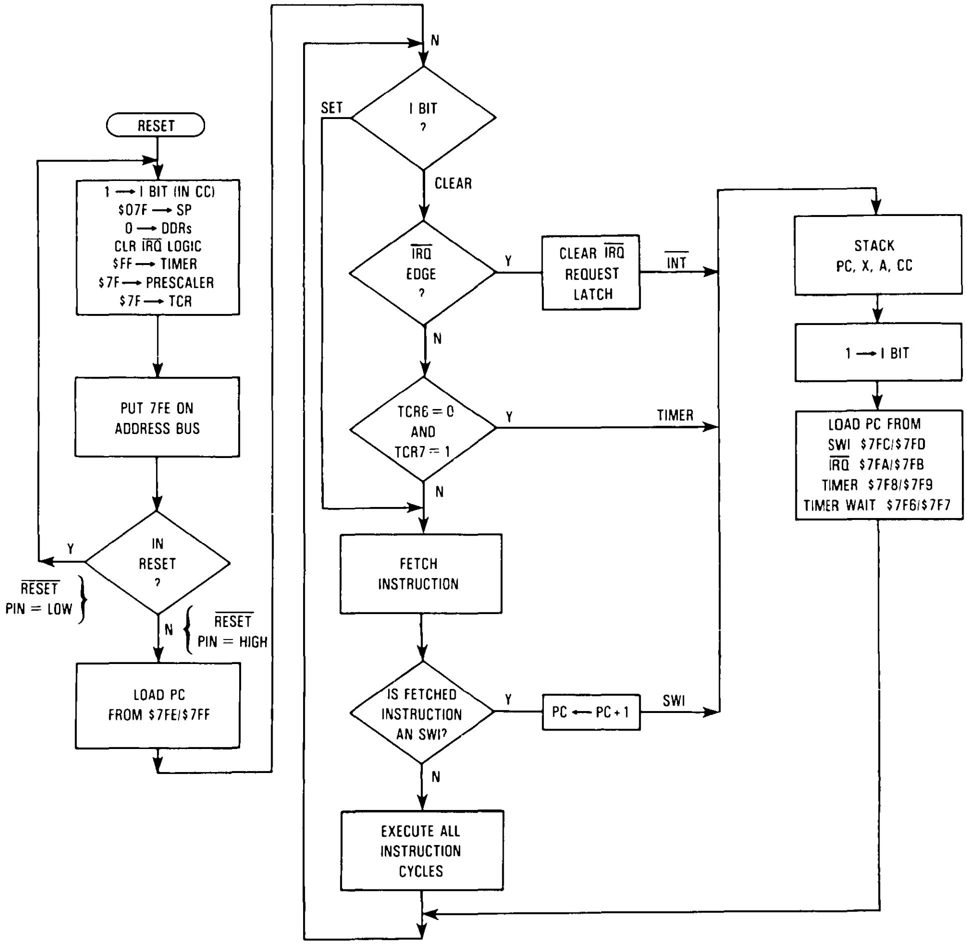

RESET
1 → I BIT (IN CC)
$07F → SP
0 → DDRs
CLR $\overline{\text{IRQ}}$ LOGIC
$FF → TIMER
$7F → PRESCALER
$7F → TCR
PUT 7FE ON ADDRESS BUS
IN RESET ?
Y
$\overline{\text{RESET}}$ PIN = LOW
N
$\overline{\text{RESET}}$ PIN = HIGH
LOAD PC FROM $7FE/$7FF
N
SET
I BIT ?
CLEAR
$\overline{\text{IRQ}}$ EDGE ?
Y
CLEAR $\overline{\text{IRQ}}$ REQUEST LATCH
$\overline{\text{INT}}$
N
TCR6 = 0 AND TCR7 = 1
Y
TIMER
N
FETCH INSTRUCTION
IS FETCHED INSTRUCTION AN SWI?
Y
PC ← PC + 1
SWI
N
EXECUTE ALL INSTRUCTION CYCLES
STACK PC, X, A, CC
1 → I BIT
LOAD PC FROM
SWI $7FC/$7FD
$\overline{\text{IRQ}}$ $7FA/$7FB
TIMER $7F8/$7F9
TIMER WAIT $7F6/$7F7

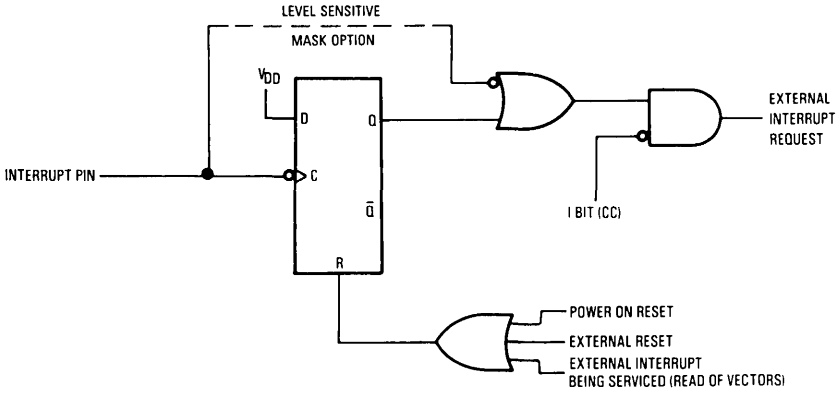
Figure 6. Reset and Interrupt Processing Flowchart

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

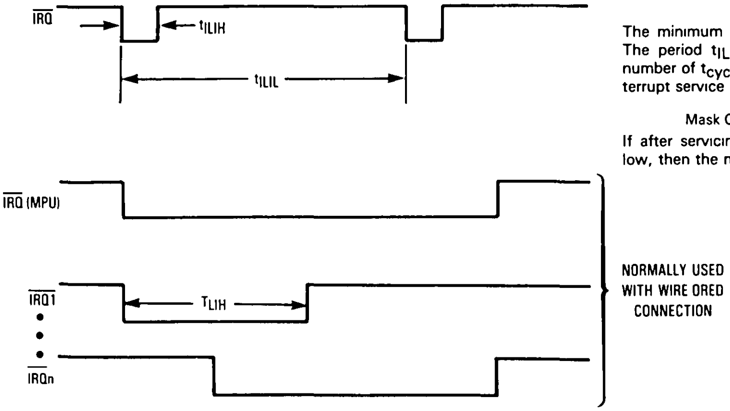
TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The various timer sources are made via the timer control register (TCR). The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 10 for timer block diagram.

(a) INTERRUPT FUNCTIONAL DIAGRAM


LEVEL SENSITIVE
MASK OPTION
VDD
D
Q
C
Q̄
R
INTERRUPT PIN
EXTERNAL
INTERRUPT
REQUEST
I BIT (CC)
POWER ON RESET
EXTERNAL RESET
EXTERNAL INTERRUPT
BEING SERVICED (READ OF VECTORS)

(b) INTERRUPT MODE DIAGRAM


IRQ
t_{ILIH}
t_{ILIL}
IRQ (MPU)
IRQ1
T_{LIH}
IRQn
NORMALLY USED
WITH WIRE ORED
CONNECTION

Edge Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles

Mask Optional Level Sensitive

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Figure 7. External Interrupt

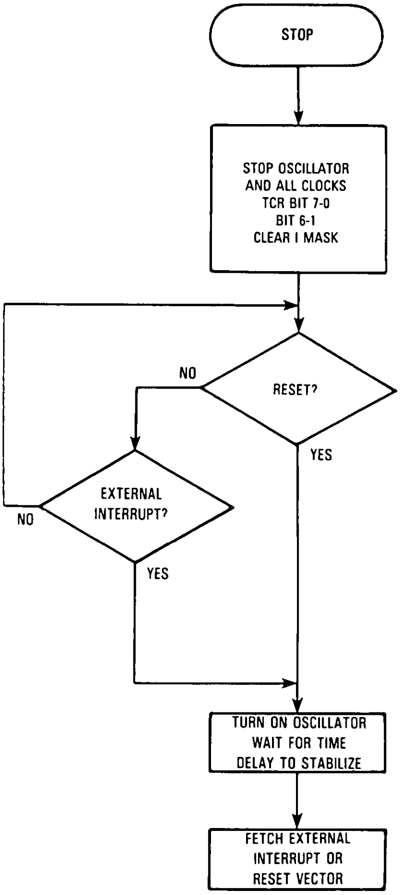

STOP
STOP OSCILLATOR
AND ALL CLOCKS
TCR BIT 7-0
BIT 6-1
CLEAR I MASK
RESET?
NO
YES
EXTERNAL
INTERRUPT?
NO
YES
TURN ON OSCILLATOR
WAIT FOR TIME
DELAY TO STABILIZE
FETCH EXTERNAL
INTERRUPT OR
RESET VECTOR

Figure 8. STOP Function Flowchart

The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared, the processor receives the interrupt. The MCU responds to this interrupt by 1) saving the present CPU state on the stack, 2) fetching the timer interrupt vector, and 3) executing the interrupt routine. The timer interrupt request bit must be cleared by software. Refer to **RESETS** and **INTERRUPTS** for additional information.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic 1; however, the TCR bit 3 always reads as a logic 0 to ensure proper operation with read-modify-write instructions.

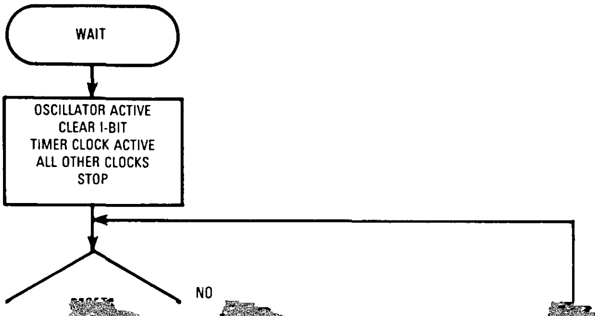
The timer continues to count past zero, falling from $00 through $FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process.

SOFTWARE CONTROLLED MODE

The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TCR4 and TCR5). The following paragraphs describe the different modes.

Timer Input Mode 1

When TCR4 and TCR5 are both programmed to zero, the timer input is from the internal clock (phase 2) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement. During the WAIT instruction, the internal clock to the timer continues to run at its normal rate.


WAIT
OSCILLATOR ACTIVE
CLEAR I-BIT
TIMER CLOCK ACTIVE
ALL OTHER CLOCKS
STOP
NO

Timer Input Mode 2

When TCR4 = 1 and TCR5 = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is ±1.

Timer Input Mode 3

When TCR4 = 0 and TCR5 = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

Timer Input Mode 4

When TCR4 and TCR5 are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts.

TIMER CONTROL REGISTER (TCR) $009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3.

	7	6	5	4	3	2	1	0
	TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
RESET	0	1	U	U	U	U	U	U

TCR7 — Timer Interrupt Request

Used to indicate the timer interrupt when it is logic one.

1 = Set when the timer data register changes to all zeros.
0 = Cleared by external reset, power-on reset, or under program control.

TCR6 — Timer Interrupt Mask

Used to inhibit the timer interrupt.

1 = Interrupt inhibited.
0 = Interrupt enabled.

TCR5 — External or Internal

Selects input clock source.

1 = External clock selected.
0 = Internal clock selected (f_{OSC}/4).

TCR4 — TIMER External Enable

Used to enable external TIMER pin or to enable the internal clock.

1 = Enables external timer pin.
0 = Disables external timer pin.

TCR3 — Prescaler Clear

Write only bit. Writing a 1 to this bit resets the prescaler to zero. A read of this location always indicates a zero.

TCR2, TCR1, TCR0 — Prescaler select bits

Decoded to select one of eight outputs of the prescaler.

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	−1
0	0	1	−2
0	1	0	−4
0	1	1	−8
1	0	0	−16
1	0	1	−32
1	1	0	−64
1	1	1	−128

INSTRUCTION SET

The MCU has a set of 61 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction listing.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following listing of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two byte instructions. Refer to the following listing of branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following listing of bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n = 0 . . . 7)
Set Bit n	BSET n (n = 0 . . . 7)
Clear Bit n	BCLR n (n = 0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following listing of control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 2 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code coversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEX, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this 2-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the direct addressing of the byte to which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports, A and B are write only registers (registers at $005 and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, these instructions cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +6.0	V
All Input Voltages except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805F2 MC146805F2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation it is recommended the V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and V_{PP} are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		115	
Cerdip		65	
Quad Package		TBD	

DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Output High Voltage ($I_{Load}=-200$ µA) PA0-PA7, PB0-PB7	V_{OH}	4.1	—	V
Output Low Voltage ($I_{Load}=800$ µA) PA0-PA7, PB0-PB7	V_{OL}	—	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	 $V_{DD}-2.0$ $V_{DD}-0.8$	 V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Total Supply Current ($C_L=50$ pF on Ports, no dc Loads, $t_{cyc}=1$ µs) RUN ($V_{IL}=0.2$ V, $V_{IH}=V_{DD}-0.2$ V) WAIT (See Note) STOP (See Note)	I_{DD}	 — — —	 4 1.5 150	 mA mA µA
I/O Ports Input Leakage — PA0-PA7, PB0-PB7	I_{IL}	—	±10	µA
Input Current — $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	I_{in}	—	±1	µA
Output Capacitance — Ports A and B	C_{out}	—	12	pF
Input Capacitance — $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	C_{in}	—	8	pF

NOTE:
Test conditions for I_{DD} are as follows:
All ports programmed as inputs
$V_{IL}=0.2$ V (PA0-PA7, PB0-PB7, PC0-PC3)
$V_{IH}=V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
OSC1 input is a square wave from 0.2 V to $V_{DD}-0.2$ V (for WAIT I_{DD} measurement only)
OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)

Table 8. Control Timing (V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0, $T_A = T_L$ to T_H, f_{osc} = 4 MHz, t_{cyc} = 1 µs)

Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (see Figure 15)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time — Crystal Oscillator (See Figure 16)	t_{ILCH}	—	100	ms
Timer Pulse Width (see Figure 14)	t_{TH}, t_{TL}	0.5	—	t_{cyc}
$\overline{\text{RESET}}$ Pulse Width (see Figure 15)	t_{RL}	1.5	—	t_{cyc}
Timer Period (see Figure 14)	t_{TLTL}	1.0	—	t_{cyc}
Interrupt Pulse Width Low (see Figure 7)	t_{ILIH}	1.0	—	t_{cyc}
Interrupt Pulse Period (see Figure 7)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	100	—	ns
Cycle Time	t_{cyc}	1000	—	ns
Frequency of Operation Crystal External Clock	f_{osc}	 — dc	 4.0 4.0	MHz

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

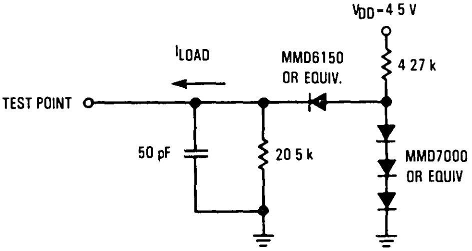


Figure 11. Equivalent Test Load

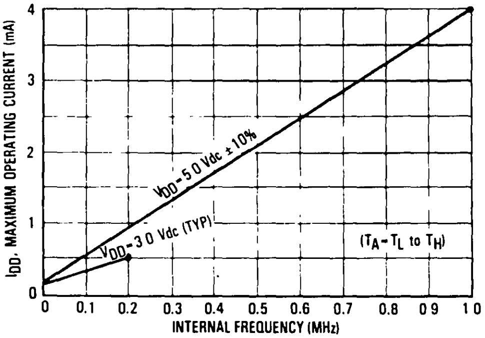


Figure 12. Maximum Operating Current vs Internal Frequency ($T_A = T_L$ to T_H)

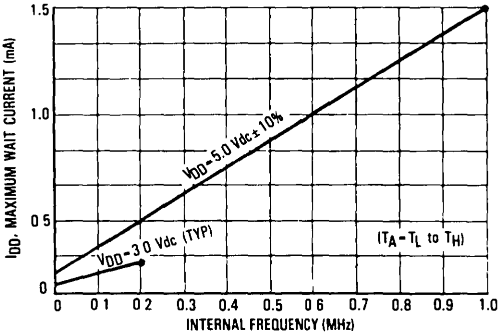


Figure 13. Maximum Wait Current vs Internal Frequency ($T_A = T_L$ to T_H)

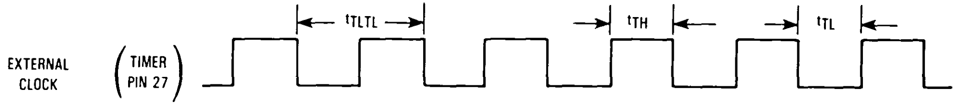
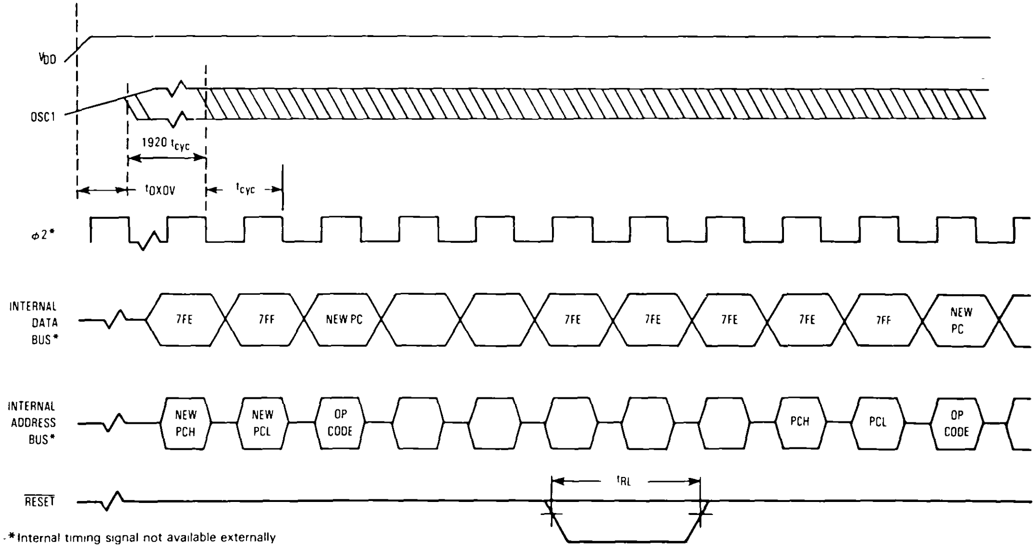
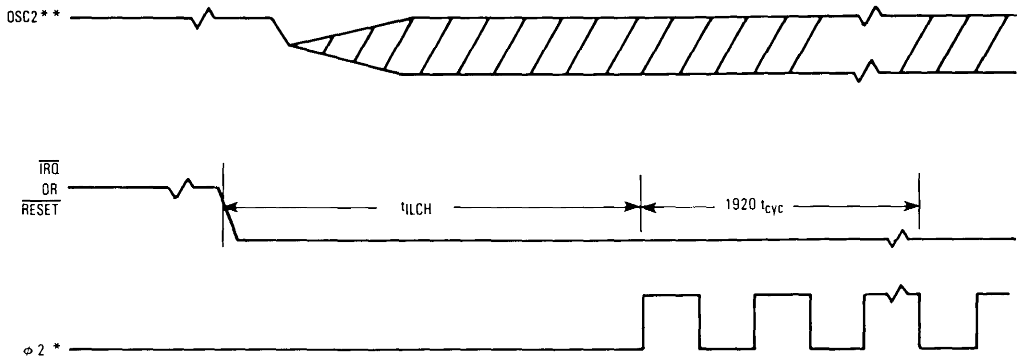

EXTERNAL CLOCK
(TIMER PIN 27)
t_{TLTL}
t_{TH}
t_{TL}

Figure 14. Timer Relationships


V_{DD}
OSC1
1920 t_{cyc}
t_{OXOV}
t_{cyc}
φ2*
INTERNAL DATA BUS*
7FE
7FF
NEW PC
7FE
7FE
7FE
7FE
7FF
NEW PC
INTERNAL ADDRESS BUS*
NEW PCH
NEW PCL
OP CODE
PCH
PCL
OP CODE
RESET
t_{RL}

*Internal timing signal not available externally

Figure 15. Power-On Reset and $\overline{\text{RESET}}$



*Internal timing signals not available externally
**Represents the internal gating of the OSC1 input pin

Figure 16. Stop Recovery

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS, disk file
MS-DOS/PC-DOS disk file
EPROM(s)MC1468705F2, 2716, or 2516

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, sales person, or Motorola representative.

NOTE

The low cost resistor oscillator option is not available on B54F mask.

FLEXIBLE DISKS

Several types of flexible disks (MDOS™ or MS™-DOS/PC-DOS disk file), programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MDOS Disk File

MDOS is Motorola's Disk Operating System available on the EXORciser® development system. The disk media submitted must be a single-sided, single density, 8-inch MDOS compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .LO output of the M6805 cross assembler should be furnished. In addition, the file must be produced (using the ROLLOUT command) containing the absolute image of the M6805 memory. It is necessary to include the entire memory image of both data and program space. All unused bytes, including those in the user space, must be set to zero.

MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A MC1468705F2, 2716, or 2516 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one of these EPROM devices, the EPROM must be programmed as described in the following paragraphs.

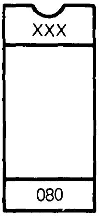
The program space ROM must start at EPROM address $080. If the customer program starts at any other address,

---

MDOS is a trademark of Motorola Inc.
MS-DOS is a trademark of Microsoft, Inc.
EXORciser® is a registered trademark of Motorola Inc.

the EPROM must be marked accordingly. All unused bytes, including the user's space, must be set at zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.

EPROM MARKING



XXX = Customer ID

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency, the MCUs are usually unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with a minimum order quantity but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

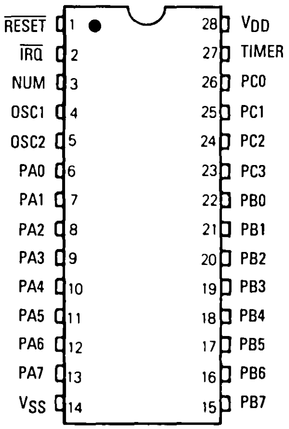
The following table provides generic information pertaining to the package type, temperature, and MC order numbers for the MC146805F2.

Table 3. Generic Information

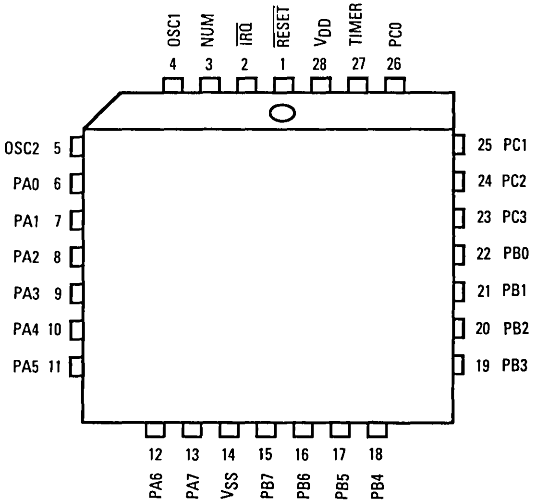
Package Type	Frequency (MHz)	Temperature	Order Number
Plastic P Suffix	4.0 4.0	0° to 70°C −40° to +85°C	MC146805F2P MC146805F2CP
PLCC FN Suffix	4.0 4.0	0° to 70°C −40° to +85°C	MC146805F2FN MC146805F2CFN
Cerdip S Suffix	4.0 4 0	0° to 70°C −40° to +85°C	MC146805F2S MC146805F2CS

PIN ASSIGNMENTS

28-Pin Dual-in-Line Package


RESET 1
IRQ 2
NUM 3
OSC1 4
OSC2 5
PA0 6
PA1 7
PA2 8
PA3 9
PA4 10
PA5 11
PA6 12
PA7 13
VSS 14
28 VDD
27 TIMER
26 PC0
25 PC1
24 PC2
23 PC3
22 PB0
21 PB1
20 PB2
19 PB3
18 PB4
17 PB5
16 PB6
15 PB7

Quad Pin Out


OSC1 4
NUM 3
IRQ 2
RESET 1
VDD 28
TIMER 27
PC0 26
OSC2 5
PA0 6
PA1 7
PA2 8
PA3 9
PA4 10
PA5 11
25 PC1
24 PC2
23 PC3
22 PB0
21 PB1
20 PB2
19 PB3
12 PA6
13 PA7
14 VSS
15 PB7
16 PB6
17 PB5
18 PB4

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146805G2

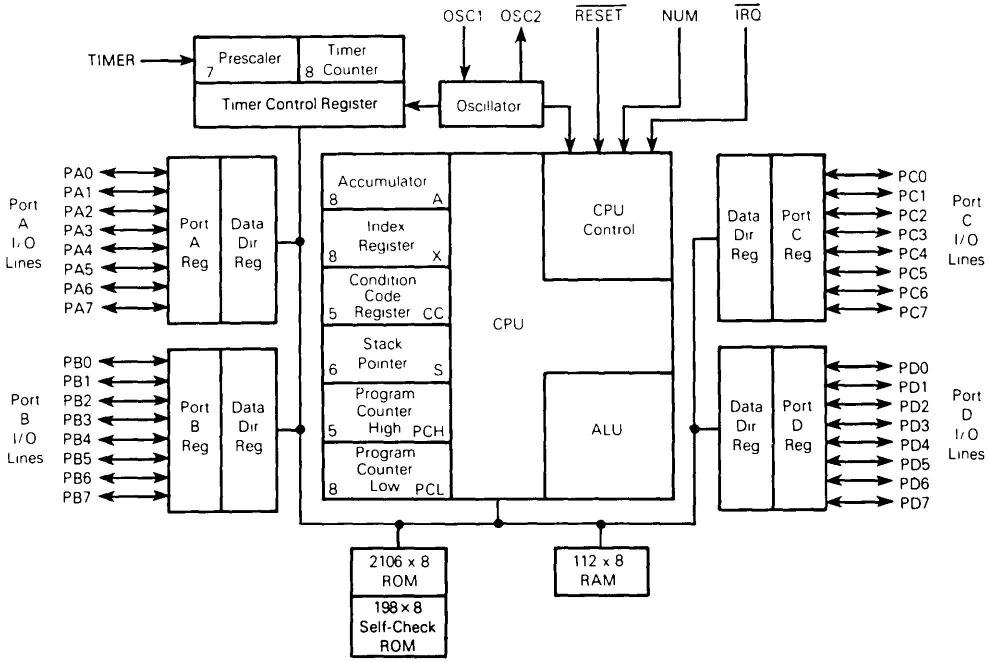
Technical Summary
8-Bit Microcontroller Unit

The MC146805G2 (CMOS) Microcomputer Unit (MCU) is a member of the MC146805 Family of microcomputers. This low cost and low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual* (M6805UM(AD2)) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the below list for additional features available on the MCU.

- Internal 8-Bit Timer with 7-Bit Programmable Prescaler
- On-chip Oscillator
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Self-Check Mode
- Power-saving STOP and WAIT Modes
- Single 3.0- to 6.0-Volt Supply
- Fully Static Operation
- 2106 Bytes ROM
- 112 Bytes RAM
- 32 I/O Ports

BLOCK DIAGRAM


OSC1
OSC2
RESET
NUM
IRQ
TIMER
Prescaler
7
Timer
8 Counter
Timer Control Register
Oscillator
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
Port A I/O Lines
Port A Reg
Data Dir Reg
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
Port B I/O Lines
Port B Reg
Data Dir Reg
Accumulator
8 A
Index Register
8 X
Condition Code
5 Register CC
Stack Pointer
6 S
Program Counter
5 High PCH
Program Counter
8 Low PCL
CPU
CPU Control
ALU
Data Dir Reg
Port C Reg
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Port C I/O Lines
Data Dir Reg
Port D Reg
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
Port D I/O Lines
2106 x 8 ROM
198 x 8 Self-Check ROM
112 x 8 RAM

This document contains information on a new product Specifications and information herein are subject to change without notice

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5Δ) power, and V_{SS} is ground.

NUM

This pin is intended for use in self-check only. In normal operation, this pin is connected to V_{SS}.

$\overline{\text{IRQ}}$

This pin is a photomask option with two different choices of interrupt triggering sensitivity: level-sensitive and negative edge-sensitive, or negative edge-sensitive only. Refer to **INTERRUPTS** for more detailed information.

OSC1, OSC2

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a ceramic resonator, or an external signal is connected to these pins to provide a system clock.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(d). This option may only be used with the crystal oscillator option selected in the mask option register. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

TIMER

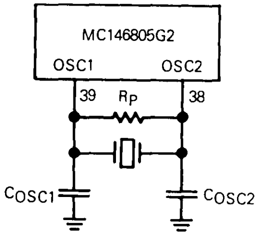
This pin is used as an external input to control the internal timer/counter circuitry.

$\overline{\text{RESET}}$

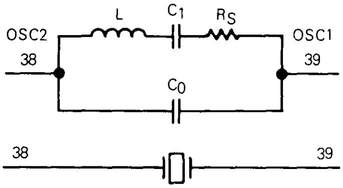
This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low.

	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

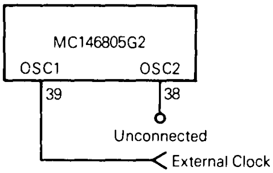
(a) Crystal Parameters



(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit



(d) External Clock Source Connections

Figure 1. Oscillator Connections.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

These 32 lines are arranged into four 8-bit ports (A B, C, and D). All ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Any port pin is programmable as either input or output under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output and a logic 0 for input. On reset, all the DDRs are initialized to a logic 0 state to put the ports in the input mode. The port output registers are not initialized on reset, and should be written to before setting the DDR bits.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and also to the latched output when the DDR is an output (1). Refer to Table 1 for I/O functions and to Figure 2 for typical port circuitry.

Table 1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers. The memory map is shown in Figure 3. The locations consist of user ROM, self-check ROM, user RAM, a Timer control register, and I/O. The interrupt vectors are located from $1FF8 to $1FFF.

The stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

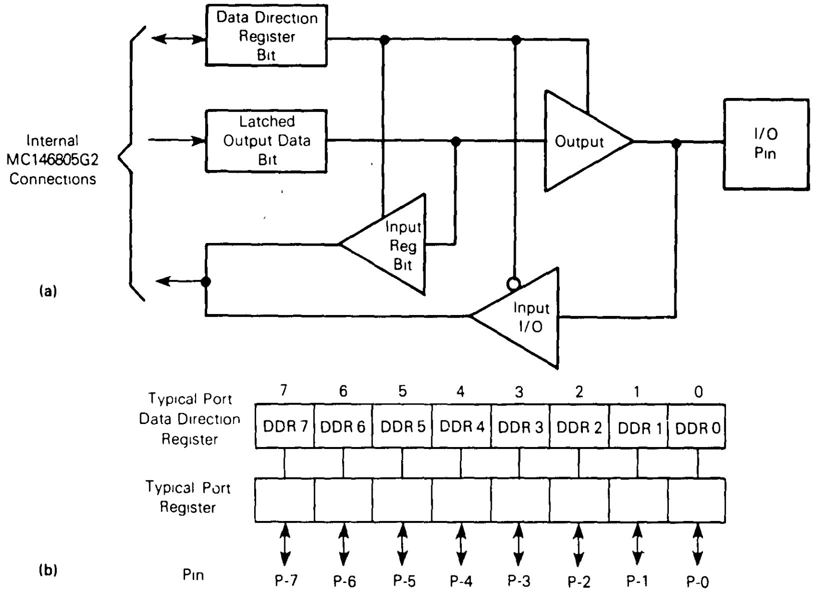
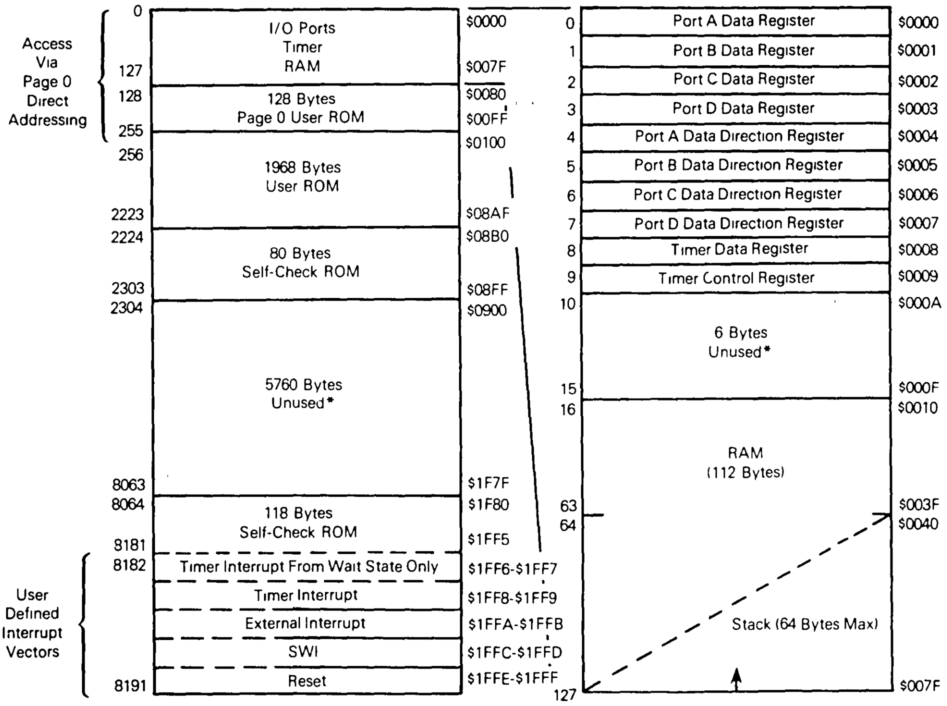


Figure 2. Typical Port I/O Circuitry and Register Configuration


Access Via Page 0 Direct Addressing
0
127
128
255
256
2223
2224
2303
2304
8063
8064
8181
8182
8191
User Defined Interrupt Vectors
I/O Ports
Timer
RAM
128 Bytes
Page 0 User ROM
1968 Bytes
User ROM
80 Bytes
Self-Check ROM
5760 Bytes
Unused*
118 Bytes
Self-Check ROM
Timer Interrupt From Wait State Only
Timer Interrupt
External Interrupt
SWI
Reset
$0000
$007F
$0080
$00FF
$0100
$08AF
$08B0
$08FF
$0900
$1F7F
$1F80
$1FF5
$1FF6-$1FF7
$1FF8-$1FF9
$1FFA-$1FFB
$1FFC-$1FFD
$1FFE-$1FFF
0
1
2
3
4
5
6
7
8
9
10
15
16
63
64
127
Port A Data Register
Port B Data Register
Port C Data Register
Port D Data Register
Port A Data Direction Register
Port B Data Direction Register
Port C Data Direction Register
Port D Data Direction Register
Timer Data Register
Timer Control Register
6 Bytes
Unused*
RAM
(112 Bytes)
Stack (64 Bytes Max)
$0000
$0001
$0002
$0003
$0004
$0005
$0006
$0007
$0008
$0009
$000A
$000F
$0010
$003F
$0040
$007F

*Reads of unused locations undefined

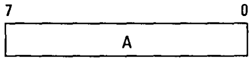
Figure 3. Memory Map

REGISTERS

The MCU contains the registers described in the following paragraphs.

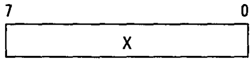
ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



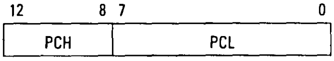
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

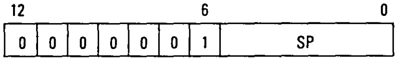
The program counter is a 13-bit register that contains the address of the next byte to be fetched.


12
8
7
0
PCH
PCL

STACK POINTER (SP)

The stack pointer is an 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The six most-significant bits of the stack pointer are permanently set at 000011. Subroutines and interrupts may be nested down to location $0040 (64 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).


12
6
0
0
0
0
0
0
0
1
SP

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

4				0
H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an external interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic 1).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. There is a 1920 t_{cyc} delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 1920 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high. At the end of POR, the timer data register contains $F0.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

INTERRUPTS

The MCU can be interrupted three different ways: (1) through the external interrupt $\overline{\text{IRQ}}$ input pin, (2) with the internal timer interrupt request, or (3) using the software interrupt instruction (SWI)

(I bit clear), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service. If the timer interrupt status bit is cleared before unmasking the interrupt, then the interrupt is not latched.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction regardless of the setting of the I bit. Refer to Figure 5 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then, each time the timer decrements to zero (transitions from $01 to $00), an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the CCR is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the condition code register is set, all interrupts are disabled. Clearing the I bit enables the external interrupts. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internallv, and

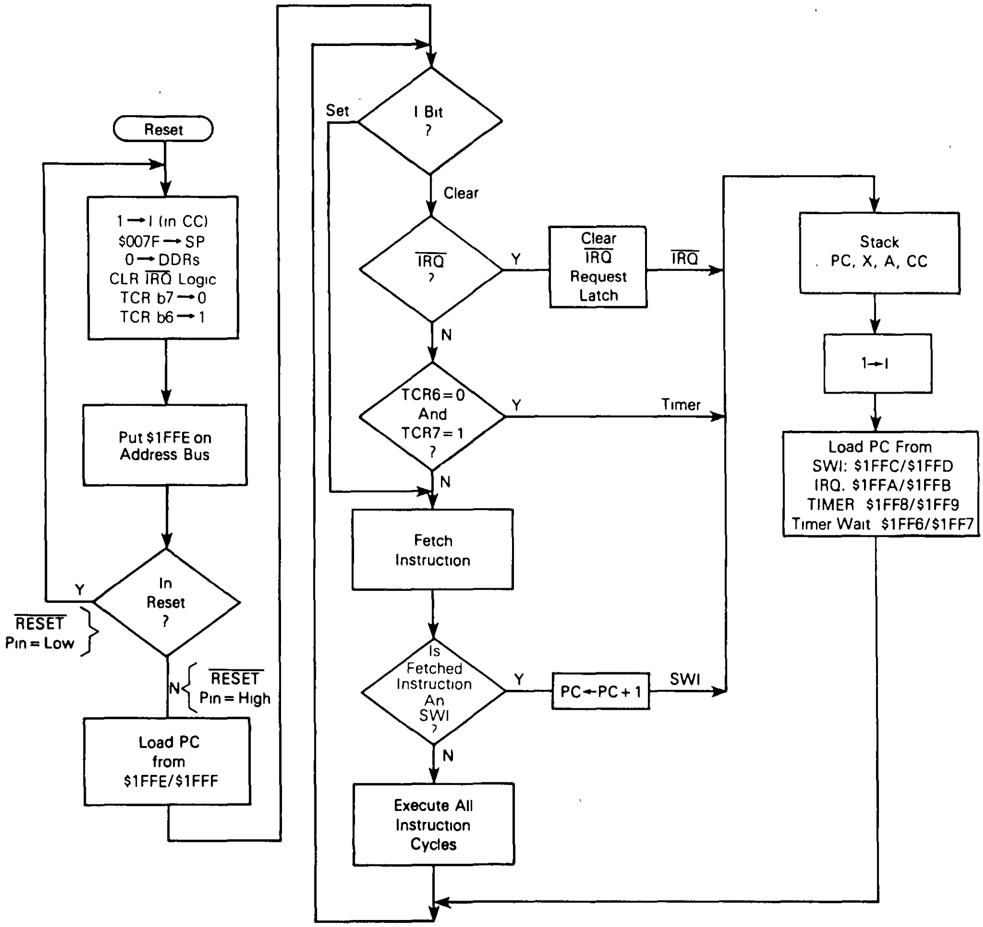

Reset
1 → I (in CC)
$007F → SP
0 → DDRs
CLR IRQ Logic
TCR b7 → 0
TCR b6 → 1
Put $1FFE on Address Bus
In Reset ?
Y
RESET Pin = Low
N
RESET Pin = High
Load PC from $1FFE/$1FFF
I Bit ?
Set
Clear
IRQ ?
Y
Clear IRQ Request Latch
IRQ
N
TCR6 = 0 And TCR7 = 1 ?
Y
Timer
N
Fetch Instruction
Is Fetched Instruction An SWI ?
Y
PC ← PC + 1
SWI
N
Execute All Instruction Cycles
Stack PC, X, A, CC
1 → I
Load PC From
SWI: $1FFC/$1FFD
IRQ. $1FFA/$1FFB
TIMER $1FF8/$1FF9
Timer Wait $1FF6/$1FF7

Figure 5. Reset and Interrupt Processing Flowchart

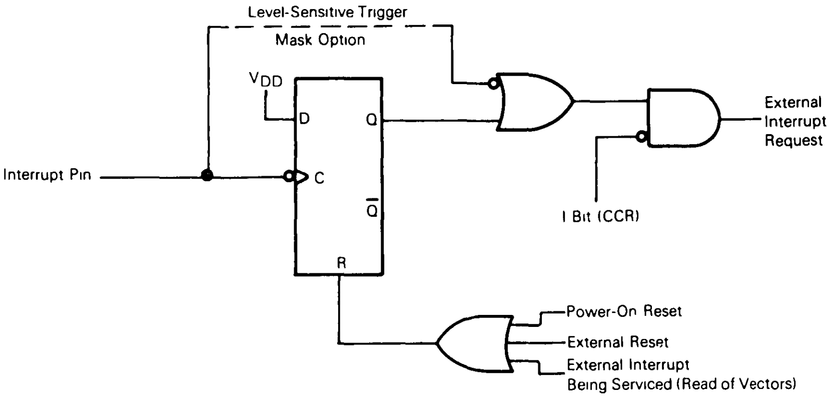
the service routine address is specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as a mask option. Figure 6 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 20 instruction cycles to the total number of cycles it takes to complete the service routine. The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

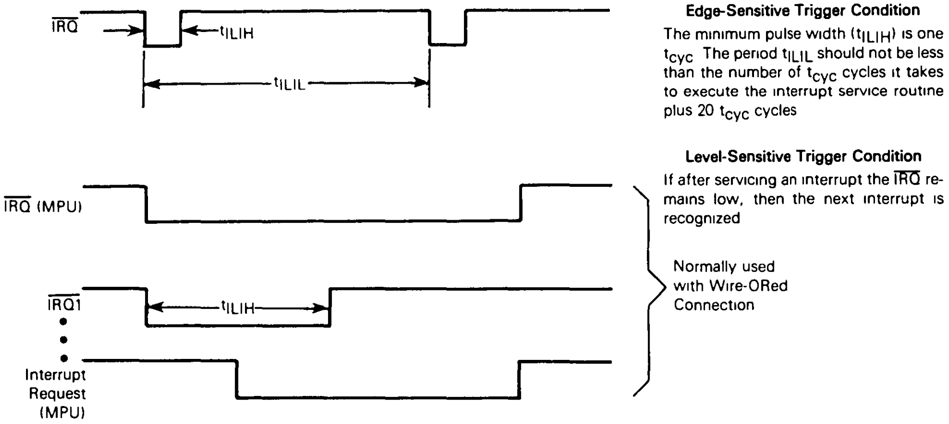
SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI execution is similar to the hardware interrupts.

(a) Interrupt Functional Diagram


Level-Sensitive Trigger
Mask Option
VDD
D
Q
C
Q̄
R
Interrupt Pin
External
Interrupt
Request
I Bit (CCR)
Power-On Reset
External Reset
External Interrupt
Being Serviced (Read of Vectors)

(b) Interrupt Mode Diagram


IRQ
tILH
tILIL
IRQ (MPU)
IRQ1
Interrupt
Request
(MPU)
tILIH
Normally used
with Wire-ORed
Connection

Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc} The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized

Figure 6. External Interrupt

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 7.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt request and to disable any further time interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can only be brought out of the STOP mode by an external interrupt or reset.

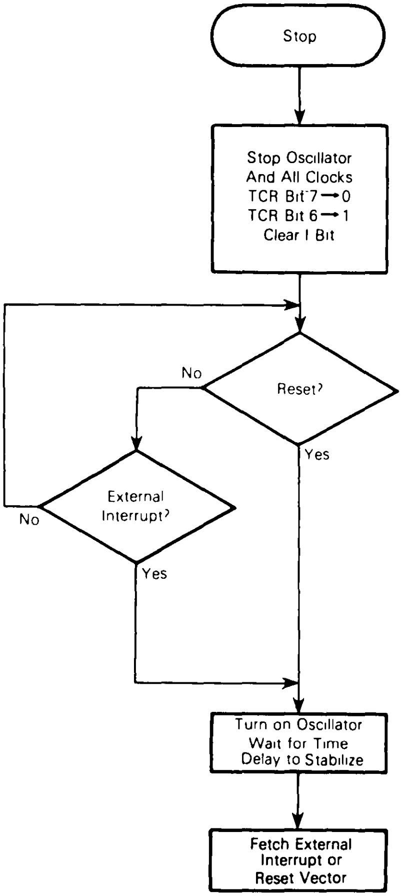

Stop
Stop Oscillator And All Clocks TCR Bit 7 → 0 TCR Bit 6 → 1 Clear I Bit
Reset?
No
Yes
External Interrupt?
No
Yes
Turn on Oscillator Wait for Time Delay to Stabilize
Fetch External Interrupt or Reset Vector

Figure 7. STOP Function Flowchart

WAIT

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 8. Thus, all internal processing is halted; however, the timer continues to count normally.

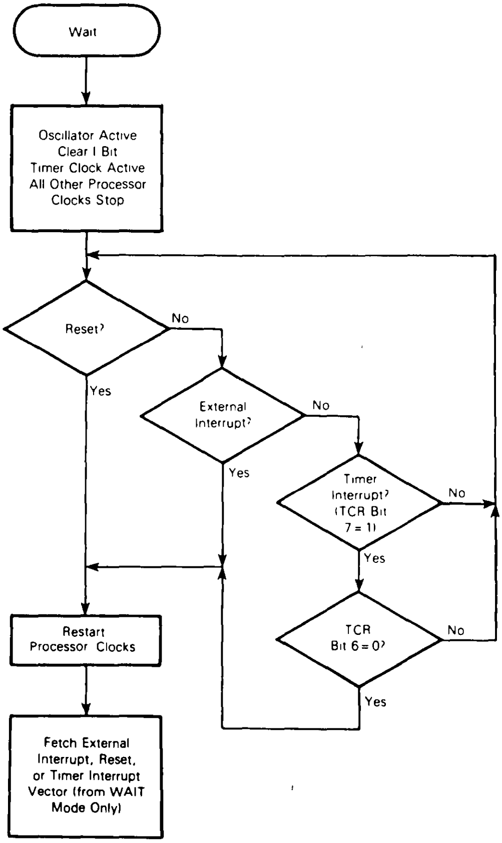

Wait
Oscillator Active Clear I Bit Timer Clock Active All Other Processor Clocks Stop
Reset?
No
Yes
External Interrupt?
No
Yes
Timer Interrupt? (TCR Bit 7 = 1)
No
Yes
TCR Bit 6 = 0?
No
Yes
Restart Processor Clocks
Fetch External Interrupt, Reset, or Timer Interrupt Vector (from WAIT Mode Only)

Figure 8. WAIT Function Flowchart

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The various timer sources are made via the timer control register (TCR). The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 9 for timer block diagram.

The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared and TCR bit 6 is cleared, the processor receives the interrupt. The MCU responds to this interrupt by 1) saving the present CPU state on the stack, 2) fetching the timer interrupt vector, and 3) executing the interrupt routine. The timer interrupt request bit must be cleared by software. Refer to **RESETS** and **INTERRUPTS** for additional information.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic 1; however, the TCR bit 3 always reads as a logic 0 to ensure proper operation with read-modify-write instructions.

The timer continues to count past zero, falling from $00 through $FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process. TDR is unaffected by reset.

SOFTWARE CONTROLLED MODE

The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TCR4 and TCR5). The following paragraphs describe the different modes.

Timer Input Mode 1

When TCR4 and TCR5 are both programmed to zero, the timer input is from the internal clock (phase 2) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement. During the WAIT instruction, the internal clock to the timer continues to run at its normal rate.

Timer Input Mode 2

When TCR4 = 1 and TCR5 = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The active high, external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is ±1.

Timer Input Mode 3

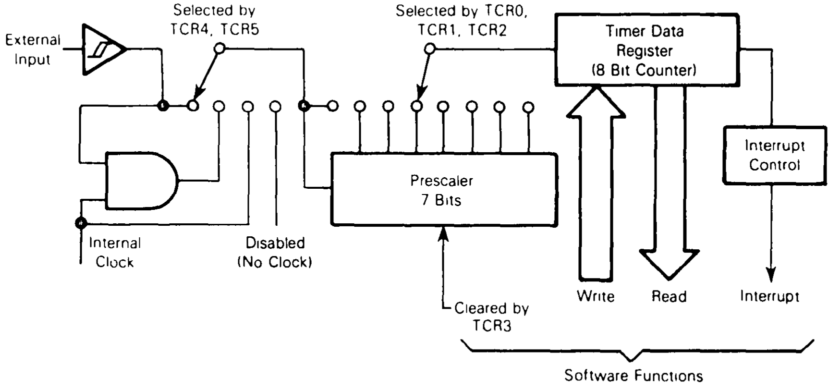
When TCR4 = 0 and TCR5 = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

Timer Input Mode 4

When TCR4 and TCR5 are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts. Power-on reset and the STOP instruction cause the counter to be set to $F0.

TIMER CONTROL REGISTER (TCR) $009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the



NOTES
1 Prescaler and timer data register (8-bit counter) are clocked on the falling edge of the internal clock or external input
2 The timer data register counts down continuously

Figure 9. Timer Block Diagram

prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3.

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

RESET CONDITION:

1	0	U	U	U	U	U	U

TCR7 — Timer Interrupt Request
- Used to indicate the timer interrupt when it is logic one
  - 1 = Set when the timer data register changes to all zeros
  - 0 = Cleared by external reset, power-on reset, or under program control

TCR6 — Timer Interrupt Mask
- Used to inhibit the timer interrupt
  - 1 = Interrupt inhibited
  - 0 = Interrupt enabled

TCR5 — External or Internal
- Selects input clock source
  - 1 = External clock selected
  - 0 = Internal clock selected ($f_{OSC}/4$)

TCR4 — TIMER External Enable
- Used to enable external TIMER pin
  - 1 = Enables external timer pin
  - 0 = Disables external timer pin

TCR3 — Prescaler Clear
- Write only bit. Writing a 1 to this bit resets the prescaler to zero. A read of this location always indicates a zero.

TCR2, TCR1, TCR0 — Prescaler Select Bits
- Decoded to select one of eight outputs of the prescaler

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following list of instructions.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch iff Higher	BHI
Branch iff Lower or Same	BLS
Branch iff Carry Clear	BCC
(Branch iff Higher or Same)	(BHS)
Branch iff Carry Set	BCS
(Branch iff Lower)	(BLO)
Branch iff Not Equal	BNE
Branch iff Equal	BEQ
Branch iff Half Carry Clear	BHCC
Branch iff Half Carry Set	BHCS
Branch iff Plus	BPL
Branch iff Minus	BMI
Branch iff Interrupt Mask Bit is Clear	BMC
Branch iff Interrupt Mask Bit is Set	BMS
Branch iff Interrupt Line is Low	BIL
Branch iff Interrupt Line is High	BIH
Branch to Subroutine	BSR

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for manipulation instructions.

Function	Mnemonic
Branch iff Bit n is Set	BRSET n (n=0 . . . 7)
Branch iff Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

OPCODE MAP SUMMARY

Table 2 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code coversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added

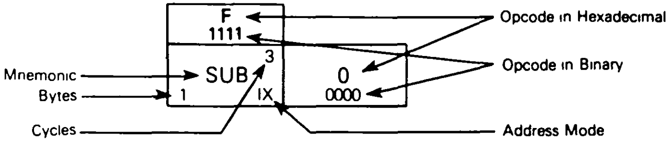
Table 2. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL								2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address.

INDEX, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the direct addressing of the byte to which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

CAUTION

The corresponding DDRs for ports, A, B, C, and D are write only registers. A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, these instructions cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit to be tested, and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD}, V_{SS}	I	10	mA
Operating Temperature Range MC146805G2 MC146805G2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain Total (PD4-PD7 only)	I_{OH}	40	mA

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation it is recommended the V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and NUM are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic PLCC	θ_{JA}	 100 70	°C/W

Port	R_1	R_2
B and C	24.3 kΩ	4.32 kΩ
A, PD0-PD3	1.21 kΩ	3.1 kΩ
PD4-PD7	300 Ω	1.64 kΩ

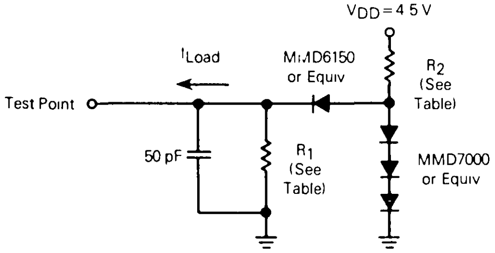


Figure 10. Equivalent Test Load

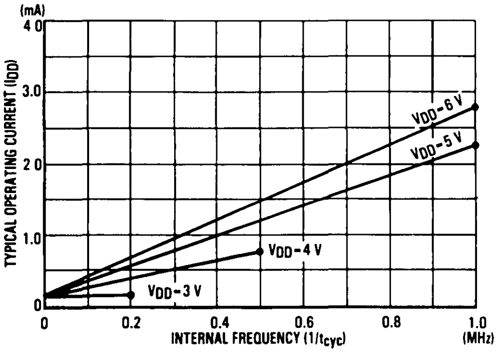


Figure 11. Typical Operating Current vs Internal Frequency

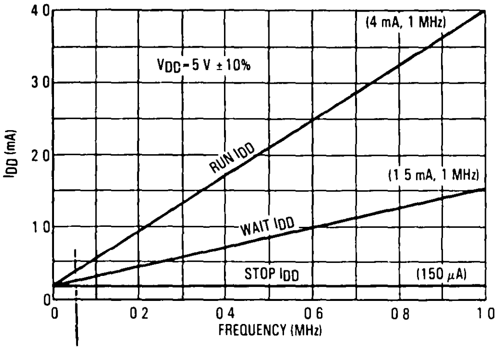


Figure 12. Maximum I_{DD} vs Frequency

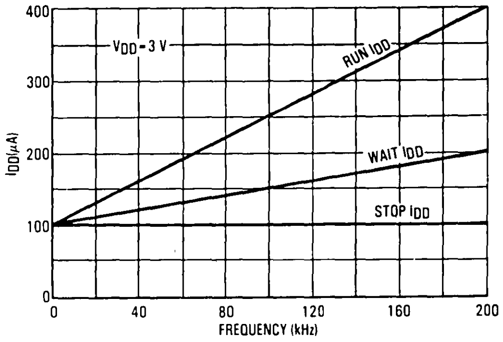


Figure 13. Maximum I_{DD} vs Frequency

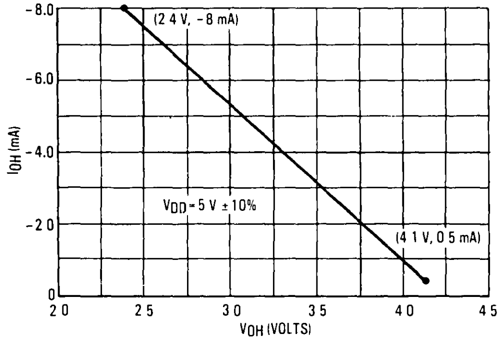


Figure 14. Minimum I_{OH}, Port D Pins 33-36

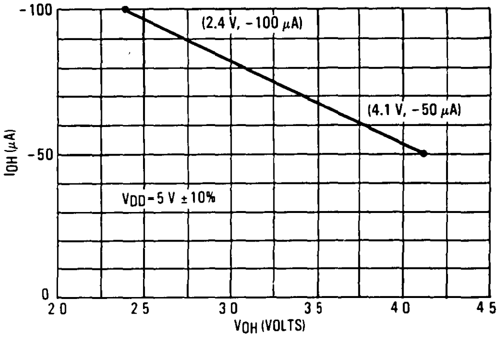


Figure 15. Minimum I_{OH}, Port B and C

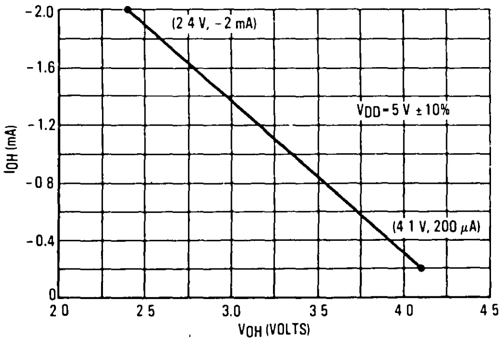


Figure 16. Minimum I_{OH}, Port A and D

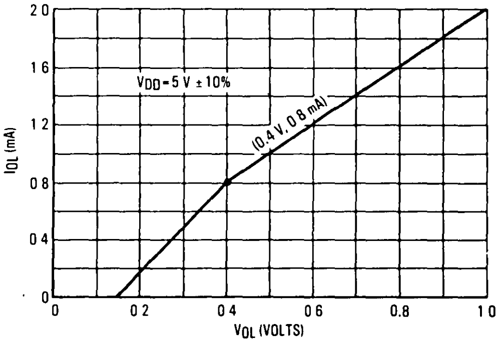


Figure 17. Minimum I_{OL}, All Ports

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Output High Voltage	V_{OH}			V
($I_{Load} = -100\ \mu A$) PB0-PB7, PC0-PC7		2.4	—	
($I_{Load} = -2$ mA) PA0-PA7, PD0-PD3		2.4	—	
($I_{Load} = -8$ mA) PD4-PD7		2.4	—	
Output Low Voltage ($I_{Load} = 800\ \mu A$) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	—	0.4	V
Input High Voltage	V_{IH}			V
Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7		$V_{DD}-2.0$	V_{DD}	
TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1		$V_{DD}-0.8$	V_{DD}	
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current ($C_L = 50$ pF on Ports, no dc Loads, $t_{cyc} = 1\ \mu s$)	I_{DD}			
RUN ($V_{IL} = 0.2$ V, $V_{IH} = V_{DD}-0.2$ V)		—	4	mA
WAIT (See Note)		—	1.5	mA
STOP (See Note)		—	150	μA
I/O Ports Input Leakage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	—	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	—	±1	μA
Capacitance				pF
Ports	C_{out}	—	12	
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	—	8	

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.0 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 1.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Output High Voltage	V_{OH}			V
($I_{Load} = -50\ \mu A$) PB0-PB7, PC0-PC7		1.4	—	
($I_{Load} = -0.5$ mA) PA0-PA7,PD0-PD3		1.4	—	
($I_{Load} = -2$ mA) PD4-PD7		1.4	—	
Output Low Voltage ($I_{Load} = 300\ \mu A$) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	—	0.3	V
Input High Voltage	V_{IH}			V
Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7		2.7	V_{DD}	
TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1		2.7	V_{DD}	
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.3	V
Total Supply Current (no dc Loads, $t_{cyc} = 5\ \mu s$)	I_{DD}			
RUN ($V_{IL} = 0.1$ V, $V_{IH} = V_{DD}-0.1$ V)		—	0.5	mA
WAIT (See Note)		—	200	μA
STOP (See Note)		—	100	μA
I/O Ports Input Leakage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	—	±5	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	—	±1	μA
Capacitance				pF
Ports	C_{out}	—	12	
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	—	8	

NOTE: Test conditions for I_{DD} are as follows:
All ports programmed as inputs
$V_{IL} = 0.2$ V (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)
$V_{IH} = V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
OSC1 input is a squarewave from 0.2 V to $V_{DD}-0.2$ V
OSC2 output load = 20 pF (wait I_{DD} is affected linearly by the OSC2 capacitance).

Table 3. Control Timing

(V_{DD}=5.0 Vdc ±10%, V_{SS}=0, T_A=0° to 70°C, f_{osc}=4 MHz)

Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (see Figure 19)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 20)	t_{ILCH}	—	100	ms
Timer Pulse Width (see Figure 18)	t_{TH}, t_{TL}	0.5	—	t_{cyc}
Reset Pulse Width (see Figure 19)	t_{RL}	1.5	—	t_{cyc}
Timer Period (see Figure 18)	t_{TLTL}	1.0	—	t_{cyc}
Interrupt Pulse Width Low (see Figure 6b)	t_{ILIH}	1.0	—	t_{cyc}
Interrupt Pulse Period (see Figure 6b)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	100	—	ns
Cycle Time	t_{cyc}	1000	—	ns
Frequency of Operation	f_{osc}			MHz
Crystal		—	4.0	
External Clock		DC	4.0	

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

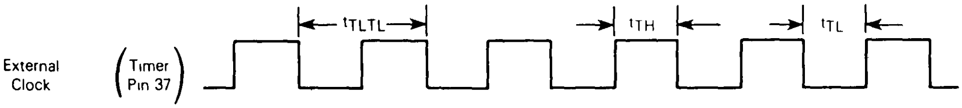
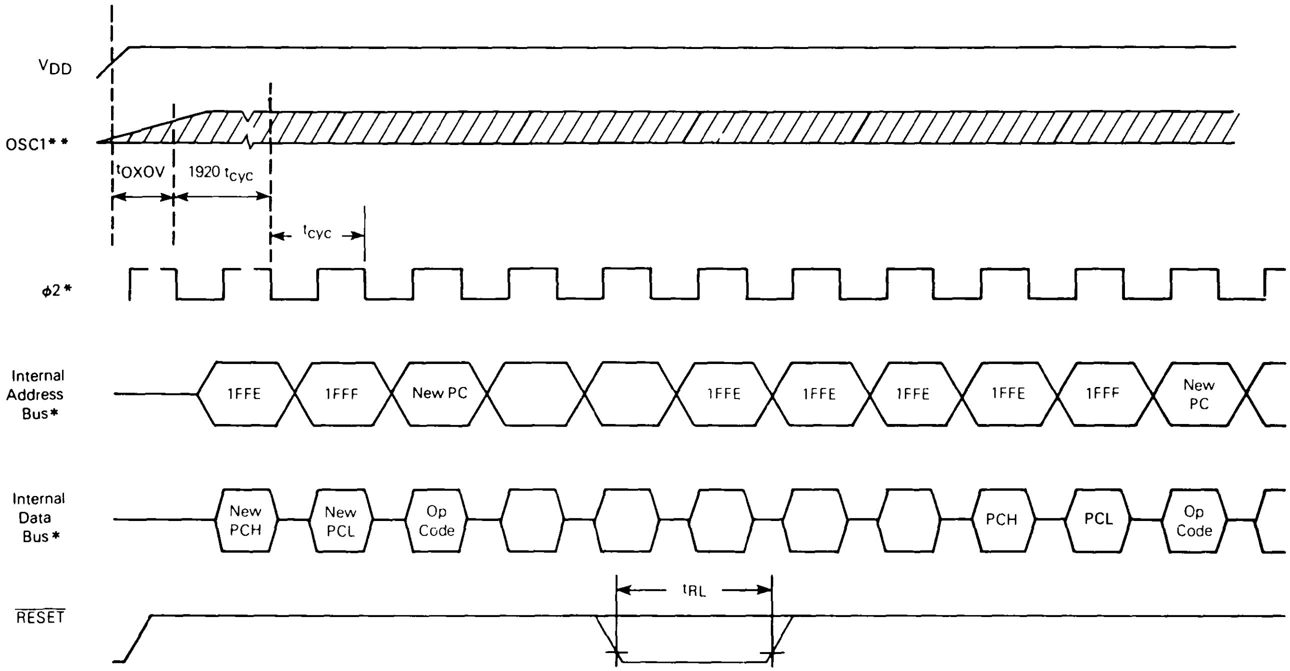


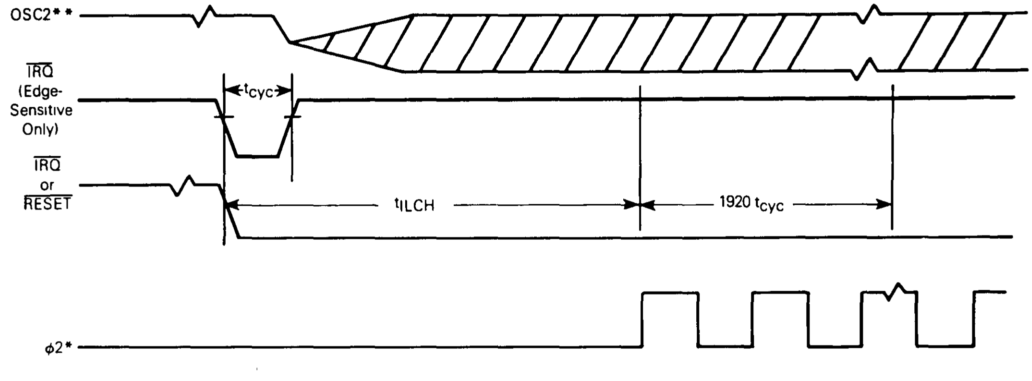
Figure 18. Timer Relationships


VDD
OSC1**
tOXOV
1920 tcyc
tcyc
φ2*
Internal Address Bus*
1FFE
1FFF
New PC
1FFE
1FFE
1FFE
1FFE
1FFF
New PC
Internal Data Bus*
New PCH
New PCL
Op Code
PCH
PCL
Op Code
tRL
RESET

*Internal timing signal and bus information not available externally

**OSC1 line is not meant to represent frequency. It is only used to represent time

Figure 19. Power-On Reset and $\overline{\text{RESET}}$



*Internal timing signals not available externally
**Represents the internal gating of the OSC1 input pin

Figure 20. Stop Recovery and Power-On Reset

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS, disk file
MS-DOS/PC-DOS disk file
EPROM(s)MC1468705G2, 2532, 2732, or two 2516/2716

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, sales person, or Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MDOS™ or MS™-DOS/PC-DOS disk file), programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MDOS DISK FILE

MDOS is Motorola's Disk Operating System available on the EXORciser® development system. The disk media submitted must be a single-sided, single density, 8-inch MDOS compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .LO output of the M6805 cross assembler should be furnished. In addition, the file must be produced (using the ROLLOUT command) containing the absolute image of the M6805 memory. It is necessary to include the entire memory image of both data and program space. All unused bytes, including those in the user space, must be set to zero.

MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

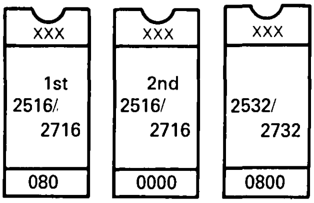
EPROMs

A MC1468705G2, 2532, 2732, 2516 (2), or 2716 (2) type EPROM(s), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one MC1468705G2/2532/2732 or two 2516/2716 type EPROM(s), the EPROM(s) must be programmed as described in the following paragraph.

For the 2532, 2732, or the 1468705G2, the ROM code should be located from $080 to $8AF and the interrupt vectors from $FF6 to $FFF. For the 2516 or 2716, the ROM code should be located from $080 to $7FF in the first EPROM and from $0 to $0AF in the second EPROM. The interrupt vectors should be in the second EPROM from $7F6 to $7FF.

MDOS is a trademark of Motorola Inc.
MS-DOS is a trademark of Microsoft, Inc.
EXORciser is a registered trademark of Motorola Inc.
IBM is a registered trademark of International Business Machines Corporation.

EPROM MARKING


XXX
1st
2516/
2716
080
XXX
2nd
2516/
2716
0000
XXX
2532/
2732
0800

XXX = Customer ID

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disk from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with 5 volts at room temperature. These RVUs are free with the minimum order quantity but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

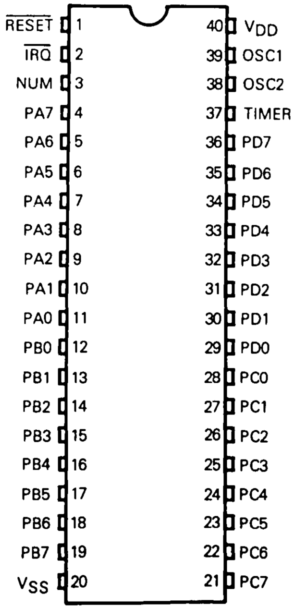
ORDERING INFORMATION

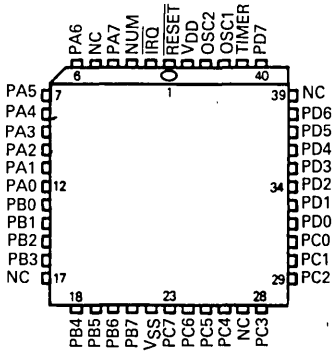
The following table provides generic information pertaining to the package type, temperature, and MC order numbers for the MC146805G2.

Table 4. Generic Information

Package Type	Frequency (MHz)	Temperature	Order Number
Plastic	1.0	0°C to 70°C	MC146805G2P
P Suffix	1.0	−40° to +85°C	MC146805G2CP
PLCC	1.0	0° to 70°C	MC146805G2FN
FN Suffix	1.0	−40° to +85°C	MC146805G2CFN

PIN ASSIGNMENTS


RESET 1
IRQ 2
NUM 3
PA7 4
PA6 5
PA5 6
PA4 7
PA3 8
PA2 9
PA1 10
PA0 11
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 18
PB7 19
VSS 20
40 VDD
39 OSC1
38 OSC2
37 TIMER
36 PD7
35 PD6
34 PD5
33 PD4
32 PD3
31 PD2
30 PD1
29 PD0
28 PC0
27 PC1
26 PC2
25 PC3
24 PC4
23 PC5
22 PC6
21 PC7


PA6
NC
PA7
NUM
IRQ
RESET
VDD
OSC2
OSC1
TIMER
PD7
6
1
40
PA5 7
PA4
PA3
PA2
PA1
PA0 12
PB0
PB1
PB2
PB3
NC 17
39 NC
PD6
PD5
PD4
PD3
34 PD2
PD1
PD0
PC0
PC1
29 PC2
18
23
28
PB4
PB5
PB6
PB7
VSS
PC7
PC6
PC5
PC4
NC
PC3

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC6809

8-Bit Microprocessing Unit

The MC6809 is a high-performance 8-bit microprocessor which supports modern programming techniques such as position independence, re-entrancy, and modular programming.

This third-generation addition to the M6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any 8-bit microprocessor.

The MC6809 has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications.

MC6800 COMPATIBLE

- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators Can Be Concatenated To Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- On-Chip Oscillator (Crystal Frequency = 4 × E)
- $\overline{\text{DMA}}/\overline{\text{BREQ}}$ Allows DMA Operation on Memory Refresh
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- MRDY Input Extends Data Access Times for Use with Slow Memory
- Interrupt Acknowledge Output Allows Vectoring by Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use with Slower Memories
- Early Write Data for Dynamic Memories

SOFTWARE FEATURES

- 10 Addressing Modes
  - — 6800 Upward Compatible Addressing Modes
  - — Direct Addressing Anywhere in Memory Map

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6809, MC68A09, MC68B09 MC6809C, MC68A09C, MC68B09C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage levels (e.g., either V_{SS} or V_{CC})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip Plastic	θ_{JA}	 60 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \tag{1}$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \tag{2}$$

Solving equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \tag{3}$$

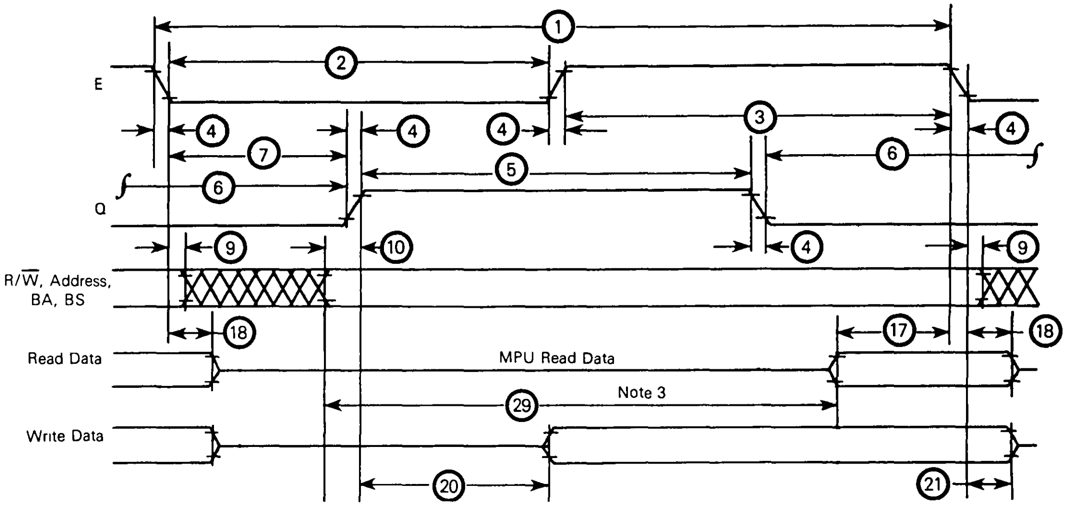
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL $\overline{\text{RESET}}$	V_{IH} V_{IHR}	V_{SS}+2.0 V_{SS}+4.0	– –	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, $\overline{\text{RESET}}$	V_{IL}	V_{SS}−0.3	–	V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic	I_{in}	–	–	2.5	μA
dc Output High Voltage (I_{Load} = −205 μA, V_{CC} = min)	D0-D7	V_{OH}	V_{SS}+2.4	–	–	V
(I_{Load} = −145 μA, V_{CC} = min)	A0-A15, R/$\overline{\text{W}}$, Q, E		V_{SS}+2.4	–	–	
(I_{Load} = −100 μA, V_{CC} = min)	BA, BS		V_{SS}+2.4	–	–	
dc Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS}+0.5	V
Internal Power Dissipation (Measured at T_A = 0°C in Steady State Operation)		P_{INT}	–	–	1.0	W
Capacitance* (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, $\overline{\text{RESET}}$	C_{in}	–	10	15	pF
	Logic Inputs, EXTAL, XTAL		–	10	15	
	A0-A15, R/$\overline{\text{W}}$, BA, BS	C_{out}	–	–	15	pF
Frequency of Operation (Crystal or External Input)	MC6809	f_{XTAL}	0.4	–	4	MHz
	MC68A09		0.4	–	6	
	MC68B09		0.4	–	8	
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = max)	D0-D7	I_{TSI}	–	2.0	10	μA
	A0-A15, R/$\overline{\text{W}}$		–	–	100	

*Capacitances are periodically tested rather than 100% tested.

FIGURE 1 — BUS TIMING



BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

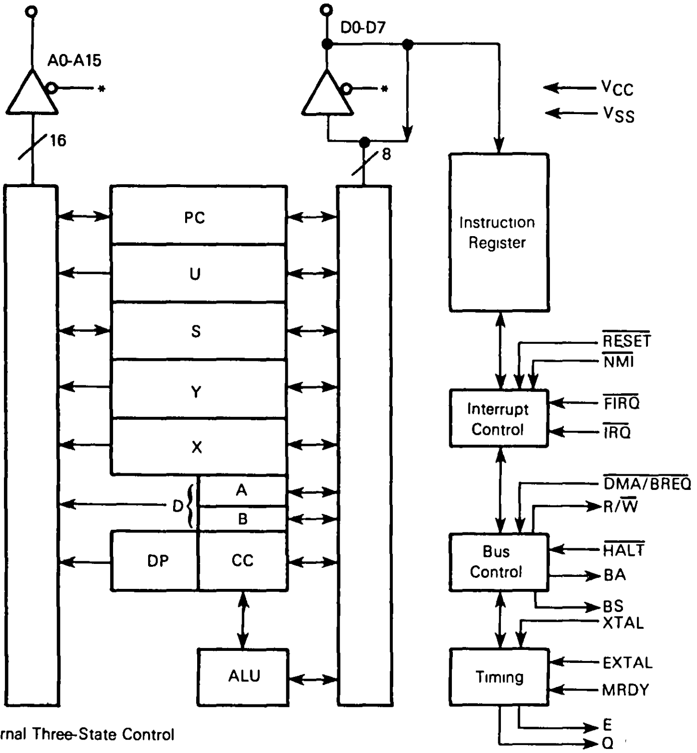
Ident. Number	Characteristics	Symbol	MC6809		MC68A09		MC68B09		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time (See Note 5)	t_{cyc}	1.0	10	0.667	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	15500	280	15700	220	15700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	430	5000	280	5000	210	5000	ns
6	Pulse Width, Q Low	PW_{QL}	450	15500	280	15700	220	15700	ns
7	Delay Time, E to Q Rise	t_{AVS}	200	250	130	165	80	125	ns
9	Address Hold Time* (See Note 4)	t_{AH}	20	–	20	–	20	–	ns
10	BA, BS, R/$\overline{W}$, and Address Valid Time to Q Rise	t_{AQ}	50	–	25	–	15	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time*	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time*	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time (See Note 3)	t_{ACC}	695	–	440	–	330	–	ns
	Processor Control Setup Time (MRDY, Interrupts, $\overline{\text{DMA/BREQ}}$, $\overline{\text{HALT}}$, $\overline{\text{RESET}}$) (Figures 6, 8, 9, 10, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	Crystal Oscillator Start Time (Figures 6 and 7)	t_{RC}	–	100	–	100	–	100	ms
	Processor Control Rise and Fall Time (Figures 6 and 8)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

*Address and data hold times are periodically tested rather than 100% tested

NOTES

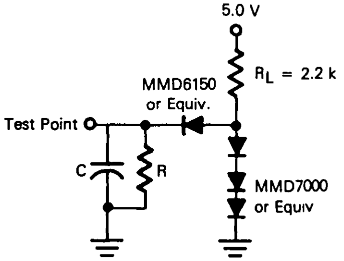
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 1 - 4 - 7 max + 10 - 17.
4. Hold time (⑨) for BA and BS is not specified
5. Maximum t_{cyc} during MRDY or $\overline{\text{DMA/BREQ}}$ is 16 µs
6. MC6809 = 1.0 MHz, MC68A09 = 1.5 MHz, MC68B09 = 2.0 MHz.

FIGURE 2 — MC6809 EXPANDED BLOCK DIAGRAM


A0-A15
D0-D7
16
8
VCC
VSS
PC
U
S
Y
X
D
A
B
DP
CC
ALU
Instruction Register
RESET
NMI
Interrupt Control
FIRQ
IRQ
DMA/BREQ
R/W
Bus Control
HALT
BA
BS
XTAL
Timing
EXTAL
MRDY
E
Q

*Internal Three-State Control

FIGURE 3 — BUS TIMING TEST LOAD


5.0 V
R_L = 2.2 k
MMD6150 or Equiv.
Test Point
C
R
MMD7000 or Equiv

C = 30 pF for BA, BS
130 pF for D0-D7, E, Q
90 pF for A0-A15, R/$\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, E, Q, R/$\overline{W}$
24 kΩ for BA, BS

PROGRAMMING MODEL

As shown in Figure 4, the MC6809 adds three registers to the set available in the MC6800. The added registers include a direct page register, the user stack pointer, and a second index register.

ACCUMULATORS (A, B, D)

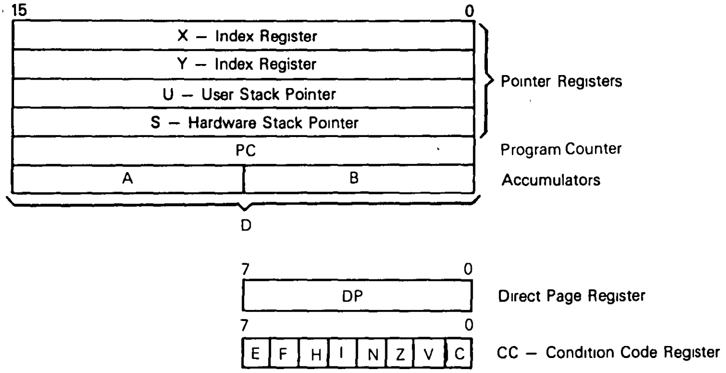
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D register, and is formed with the A register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The direct page register of the MC6809 serves to enhance the direct addressing mode. The content of this register appears at the higher address outputs (A8-A15) during direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure M6800 compatibility, all bits of this register are cleared during processor reset.

FIGURE 4 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT


15
0
X — Index Register
Y — Index Register
U — User Stack Pointer
S — Hardware Stack Pointer
Pointer Registers
PC
Program Counter
A
B
Accumulators
D
7
0
DP
Direct Page Register
7
0
E F H I N Z V C
CC — Condition Code Register

INDEX REGISTERS (X, Y)

The index registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculation of effective addresses. This address may be used to point to data directly or may be modified by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented or decremented to point to the next item of tabular type data. All four pointer registers (X, Y, U, S) may be used as index registers.

STACK POINTER (U,S)

The hardware stack pointer (S) is used automatically by the processor during subroutine calls and interrupts. The stack pointers of the MC6809 point to the top of the stack, in contrast to the MC6800 stack pointer, which pointed to the next free location on the stack. The user stack pointer (U) is controlled exclusively by the programmer. This allows arguments to be passed to and from subroutines with ease. Both stack pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support **Push** and **Pull** instructions. This allows the MC6809 to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

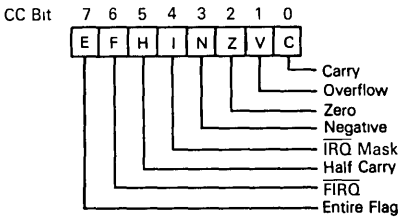
PROGRAM COUNTER

The program counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative addressing is provided allowing the program counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The condition code register defines the state of the processor at any given time. See Figure 5.

FIGURE 5 — CONDITION CODE REGISTER FORMAT


CC Bit
7 6 5 4 3 2 1 0
E F H I N Z V C
Carry
Overflow
Zero
Negative
$\overline{\text{IRQ}}$ Mask
Half Carry
$\overline{\text{FIRQ}}$
Entire Flag

CONDITION CODE REGISTER DESCRIPTION

CARRY FLAG (C)

Bit 0 is the carry flag, and is usually the carry from the binary ALU. C is also used to represent a 'borrow' from subtract-like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

OVERFLOW FLAG (V)

Bit 1 is the overflow flag, and is set to a one by an operation which causes a signed twos complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

ZERO FLAG (Z)

Bit 2 is the zero flag, and is set to a one if the result of the previous operation was identically zero.

NEGATIVE FLAG (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation. Thus, a negative twos-complement result will leave N set to a one.

$\overline{\text{IRQ}}$ MASK (I)

Bit 4 is the $\overline{\text{IRQ}}$ mask bit. The processor will not recognize interrupts from the $\overline{\text{IRQ}}$ line if this bit is set to a one. $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, and SWI all set I to a one. SWI2 and SWI3 do not affect I.

HALF CARRY (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD). This bit is used by the DAA instruction to perform a BCD decimal add adjust operation. The state of this flag is undefined in all subtract-like instructions.

$\overline{\text{FIRQ}}$ MASK (F)

Bit 6 is the $\overline{\text{FIRQ}}$ mask bit. The processor will not recognize interrupts from the $\overline{\text{FIRQ}}$ line if this bit is a one. $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, SWI, and $\overline{\text{RESET}}$ all set F to a one. $\overline{\text{IRQ}}$, SWI2, and SWI3 do not affect F.

ENTIRE FLAG (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the condition code register represents past action.

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part: V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the address bus. When the processor does not require the bus for a data transfer, it will output address $FFFF_{16}$, R/$\overline{\text{W}}$ = 1, and BS = 0; this is a "dummy access" or $\overline{\text{VMA}}$ cycle. Addresses are valid on the rising edge of Q. All address bus drivers are made high impedance when output bus available (BA) is high. Each pin will drive one Schottky TTL load or four LSTTL loads, and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus. Each pin will drive one Schottky TTL load or four LSTTL loads, and 130 pF.

READ/WRITE (R/$\overline{\text{W}}$)

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. R/$\overline{\text{W}}$ is made high impedance when BA is high. R/$\overline{\text{W}}$ is valid on the rising edge of Q.

$\overline{\text{RESET}}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 6. The reset vectors are fetched from locations $FFFE_{16}$ and $FFFF_{16}$ (Table 1) when interrpt acknowledge is true, ($\overline{\text{BA}\bullet\text{BS}}$ = 1). During initial power on, the $\overline{\text{RESET}}$ line should be held low until the clock oscillator is fully operational. See Figure 7.

Because the $\overline{\text{RESET}}$ pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system This higher threshold voltage ensures that all peripherals are out of the reset state before the processor

$\overline{\text{HALT}}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the halt or bus grant state. While halted, the MPU will not respond to external real-time requests ($\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$) although $\overline{\text{DMA/BREQ}}$ will always be accepted, and $\overline{\text{NMI}}$ or $\overline{\text{RESET}}$ will be latched for later response. During the halt state, Q and E continue to run normally If the MPU is not runnig, ($\overline{\text{RESET}}$, $\overline{\text{DMA/BREQ}}$), a halted state (BA•BS = 1) can be achieved by pulling $\overline{\text{HALT}}$ low while $\overline{\text{RESET}}$ is still low. If $\overline{\text{DMA/BREQ}}$ and $\overline{\text{HALT}}$ are both pulled low, the processor will reach the last cycle of the instruction (by reverse cycle stealing) where the machine will then become halted See Figure 8.

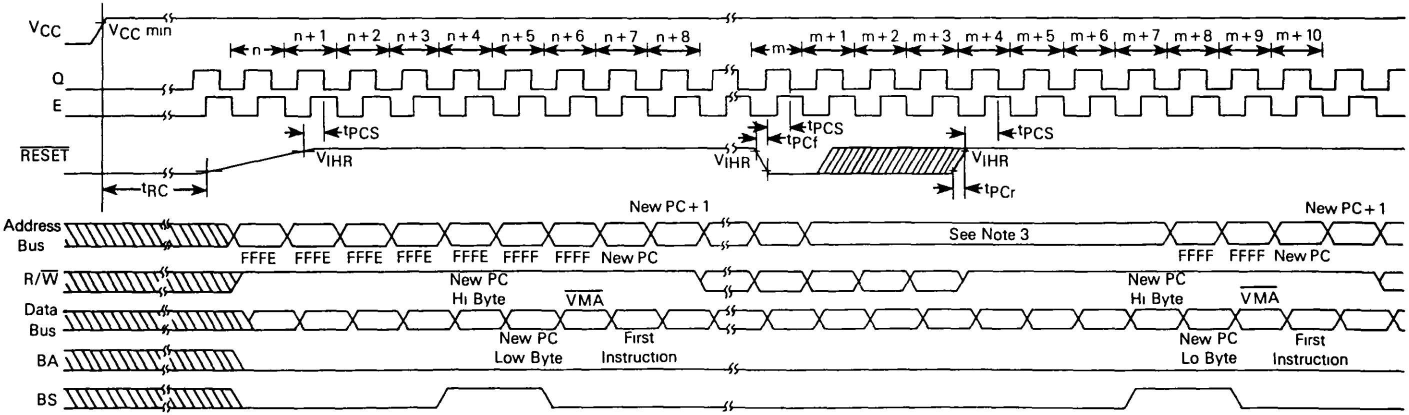
BUS AVAILABLE, BUS STATUS (BA, BS)

The bus available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. This signal does not imply that the bus will be available for more than one cycle. When BA goes low, a dead cycle will elapse before the MPU acquires the bus

The bus status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q)

MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt or Bus Grant Acknowledge

FIGURE 6 — $\overline{\text{RESET}}$ TIMING

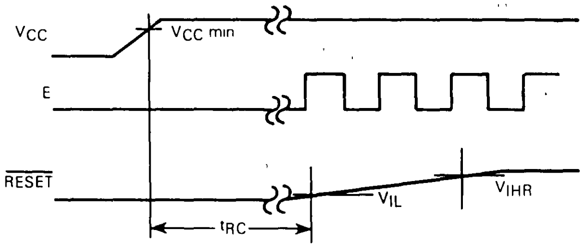

VCC
VCC min
n
n+1
n+2
n+3
n+4
n+5
n+6
n+7
n+8
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
m+10
Q
E
tPCS
tPCf
RESET
VIHR
tRC
tPCr
Address Bus
New PC+1
See Note 3
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
FFFF
New PC
FFFF
FFFF
New PC
R/W
New PC Hi Byte
VMA
Data Bus
New PC Low Byte
First Instruction
New PC Lo Byte
BA
BS

NOTES: 1. Parts with date codes prefixed by 7F or 5A will come out of $\overline{\text{RESET}}$ one cycle sooner than shown

2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2 0 volts, unless otherwise noted

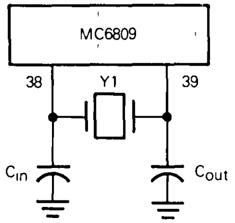
3. FFFE appears on the bus during $\overline{\text{RESET}}$ low time Following the active transition of the $\overline{\text{RESET}}$ line, three more FFFE cycles will appear followed by the vector fetch

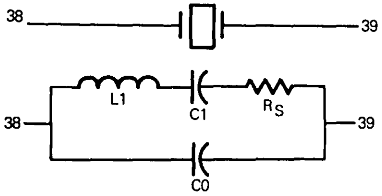
FIGURE 7 — CRYSTAL CONNECTIONS AND OSCILLATOR START UP


V_{CC}
V_{CC} min
E
$\overline{\text{RESET}}$
V_{IL}
V_{IHR}
t_{RC}

NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified

Y1	C_{in}	C_{out}
8 MHz	18 pF	18 pF
6 MHz	20 pF	20 pF
4 MHz	24 pF	24 pF


MC6809
38
Y1
39
C_{in}
C_{out}


38
39
38
L1
C1
R_S
39
C0

Nominal Crystal Parameters

	3.58 MHz	4.00 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C0	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40 k	>30 k	>20 k	>20 k

All parameters are 10%

NOTE: These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used

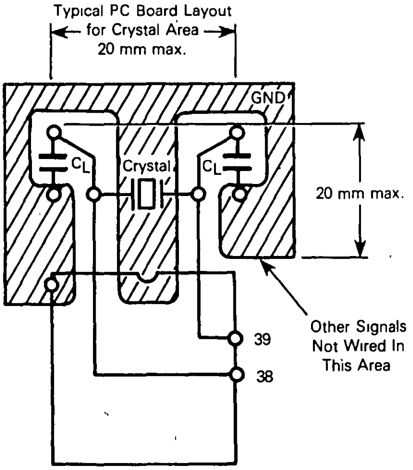
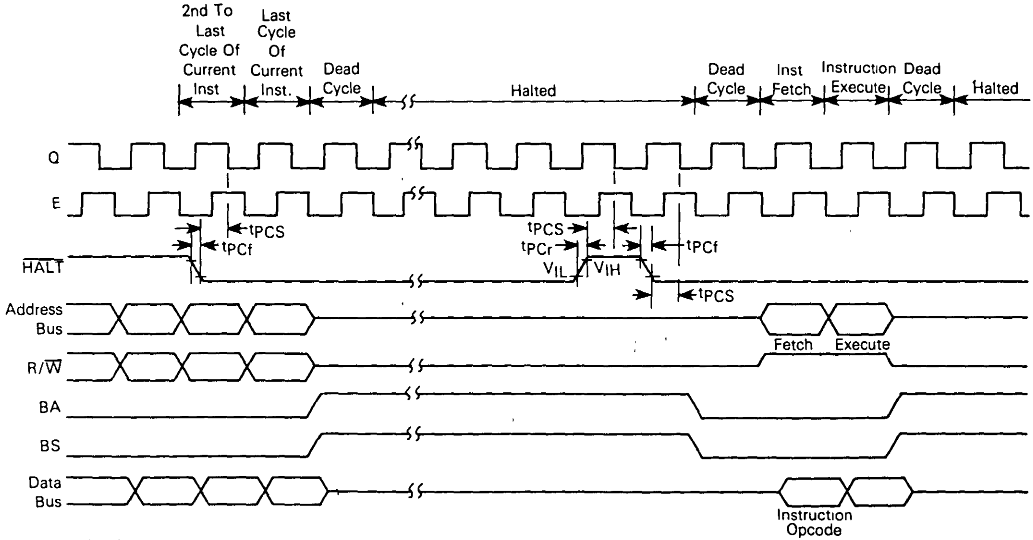

Typical PC Board Layout
for Crystal Area
20 mm max.
GND
C_L
Crystal
C_L
20 mm max.
39
38
Other Signals
Not Wired In
This Area

FIGURE 8 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



NOTE Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0 8 V unless otherwise specified

INTERRUPT ACKNOWLEDGE is indicated during both cycles of a hardware-vector-fetch ($\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, SWI, SWI2, SWI3). This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device See Table 1

SYNC ACKNOWLEDGE is indicated while the MPU is waiting for external synchronization on an interrupt line.

HALT/BUS GRANT is true when the MC6809 is in a halt or bus grant condition

TABLE 1 — MEMORY MAP FOR INTERRUPT VECTORS

Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{\text{IRQ}}$
FFF6	FFF7	$\overline{\text{FIRQ}}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

NON MASKABLE INTERRUPT ($\overline{\text{NMI}}$)*

A negative transition on this input requests that a non-maskable interrupt sequence be generated A non-maskable interrupt cannot be inhibited by the program. It also has a higher priority than $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, or software interrupts. During recognition of an $\overline{\text{NMI}}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{\text{NMI}}$ will not be recognized until the first program load of the hardware stack pointer (S). The pulse width of $\overline{\text{NMI}}$ low must be at least one E cycle. If the $\overline{\text{NMI}}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 9.

FAST-INTERRUPT REQUEST ($\overline{\text{FIRQ}}$)*

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard interrupt request ($\overline{\text{IRQ}}$), and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI See Figure 10.

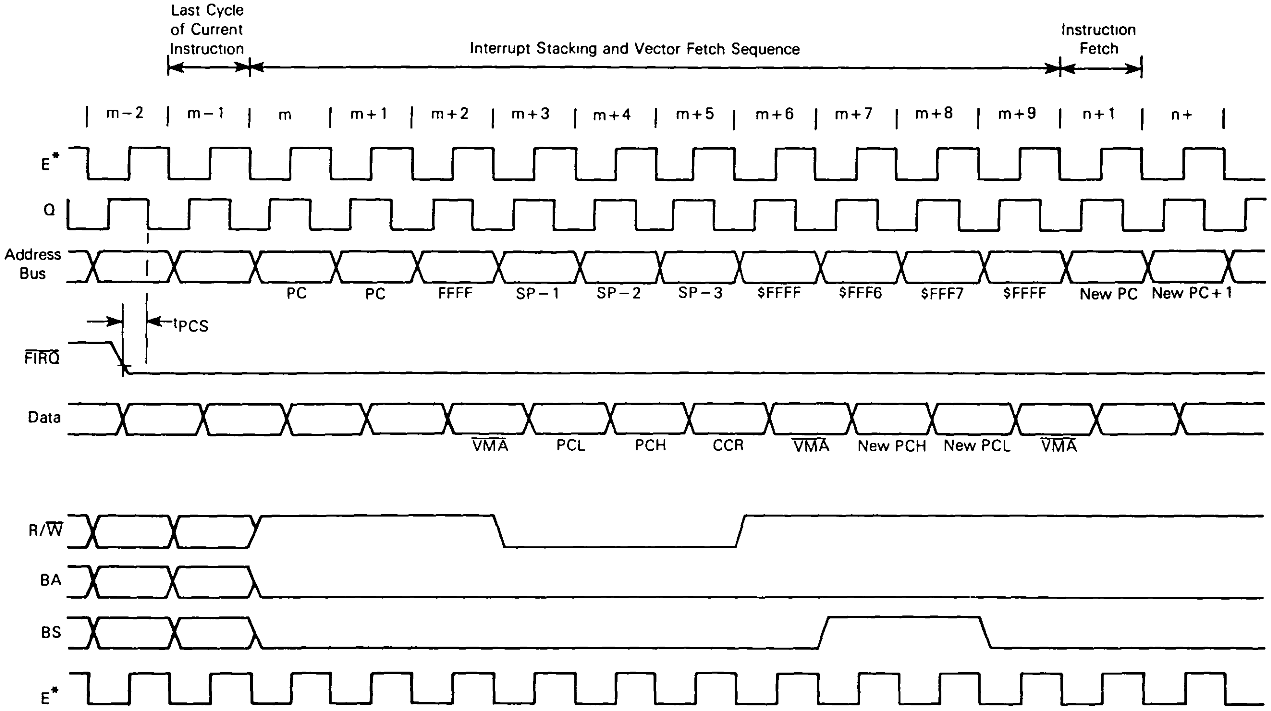
INTERRUPT REQUEST ($\overline{\text{IRQ}}$)*

A low level input on this pin will initiate an interrupt request sequence provided the mask bit (I) in the CC is clear. Since $\overline{\text{IRQ}}$ stacks the entire machine state, it provides a slower response to interrupts than $\overline{\text{FIRQ}}$. $\overline{\text{IRQ}}$ also has a lower priority than $\overline{\text{FIRQ}}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 9.

---

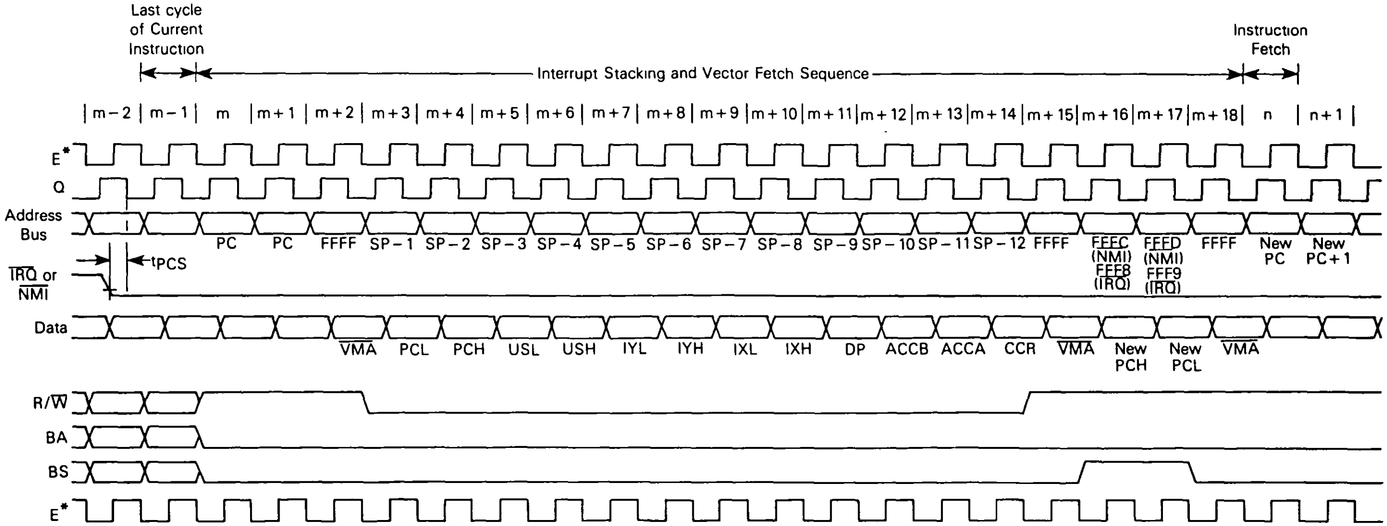
*$\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, and $\overline{\text{IRQ}}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or $\overline{\text{CWAI}}$ condition is present. If $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ do not remain low until completion of the current instruction they may not be recognized. However, $\overline{\text{NMI}}$ is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{\text{RESET}}$ and the rising edge of BS indicating $\overline{\text{RESET}}$ acknowledge.

FIGURE 10 — $\overline{\text{FIRQ}}$ INTERRUPT TIMING


Last Cycle of Current Instruction
Interrupt Stacking and Vector Fetch Sequence
Instruction Fetch
m−2
m−1
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
n+1
n+
E*
Q
Address Bus
PC
PC
FFFF
SP−1
SP−2
SP−3
$FFFF
$FFF6
$FFF7
$FFFF
New PC
New PC+1
tPCS
FIRQ
Data
VMA
PCL
PCH
CCR
VMA
New PCH
New PCL
VMA
R/W
BA
BS
E*

NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2 0 V and logic low = 0 8 V unless otherwise specified
*E clock shown for reference only

FIGURE 9 — $\overline{\text{IRQ}}$ AND $\overline{\text{NMI}}$ INTERRUPT TIMING


Last cycle of Current Instruction
Interrupt Stacking and Vector Fetch Sequence
Instruction Fetch
m−2
m−1
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
m+10
m+11
m+12
m+13
m+14
m+15
m+16
m+17
m+18
n
n+1
E*
Q
Address Bus
PC
PC
FFFF
SP−1
SP−2
SP−3
SP−4
SP−5
SP−6
SP−7
SP−8
SP−9
SP−10
SP−11
SP−12
FFFF
FFFC (NMI) FFF8 (IRQ)
FFFD (NMI) FFF9 (IRQ)
FFFF
New PC
New PC+1
tPCS
IRQ or NMI
Data
VMA
PCL
PCH
USL
USH
IYL
IYH
IXL
IXH
DP
ACCB
ACCA
CCR
VMA
New PCH
New PCL
VMA
R/W
BA
BS
E*

NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2.0 V and logic low = 0.8 V unless otherwise specified

*E clock shown for reference only

XTAL, EXTAL

These inputs are used to connect the on-chip oscillator to an external parallel-resonant crystal. Alternately, the pin EXTAL may be used as a TTL level input for external timing by grounding XTAL. The crystal or external frequency is four times the bus frequency. See Figure 7. Proper RF layout techniques should be observed in the layout of printed circuit boards

E, Q

E is similar to the MC6800 bus timing signal phase 2; Q is a quadrature clock signal which leads E. Q has no parrallel on the MC6800 Addresses from the MPU will be valid with the leading edge of Q. Data is latched on the falling edge of E Timing for E and Q is shown in Figure 11.

MRDY

This input control signal allows stretching of E and Q to extend data-access time E and Q operate normally while MRDY is high. When MRDY is low, E and Q may be stretched in integral multiples of quarter (¼) bus cycles, thus allowing interface to slow memories, as shown in Figure 12(a) During non-valid memory access ($\overline{\text{VMA}}$ cycles), MRDY has no effect on stretching E and Q; this inhibits slowing the processor during "don't care" bus accesses. MRDY may also be used to stretch clocks (for slow memory) when bus control has been transferred to an external device (through the use of $\overline{\text{HALT}}$ and $\overline{\text{DMA/BREQ}}$).

$\overline{\text{DMA/BREQ}}$

The $\overline{\text{DMA/BREQ}}$ input provides a method of suspending execution and acquiring the MPU bus for another use, as shown in Figure 13 Typical uses include DMA and dynamic memory refresh.

A low level on this pin will stop instruction execution at the end of the current cycle unless pre-empted by self-refresh. The MPU will acknowledge $\overline{\text{DMA/BREQ}}$ by setting BA and BS to a one The requesting device will now have up to 15 bus cycles before the MPU retrieves the bus for self-refresh Self-refresh requires one bus cycle with a leading and trailing dead cycle. See Figure 14. The self-refresh counter is only cleared if $\overline{\text{DMA/BREQ}}$ is inactive for two or more MPU cycles.

Typically, the DMA controller will request to use the bus by asserting $\overline{\text{DMA/BREQ}}$ pin low on the leading edge of E When the MPU replies by setting BA and BS to a one, that cycle will be a dead cycle used to transfer bus mastership to the DMA controller

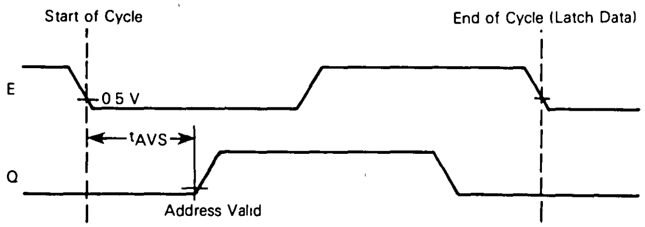
False memory accesses may be prevented during any dead cycles by developing a system $\overline{\text{DMAVMA}}$ signal which is LOW in any cycle when BA has changed

When BA goes low (either as a result of $\overline{\text{DMA/BREQ}}$ HIGH or MPU self-refresh), the DMA device should be taken off the bus. Another dead cycle will elapse before the MPU accesses memory to allow transfer of bus mastership without contention.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function This sequence begins after $\overline{\text{RESET}}$ and is repeated indefinitely unless altered by a special instruction or hardware occurrence Software instructions that alter normal MPU operation are SWI, SWI2, SWI3, CWAI, RTI, and SYNC. An interrupt, $\overline{\text{HALT}}$, or $\overline{\text{DMA/BREQ}}$ can also alter the normal execution of instructions Figure 15 is the flowchart for the MC6809

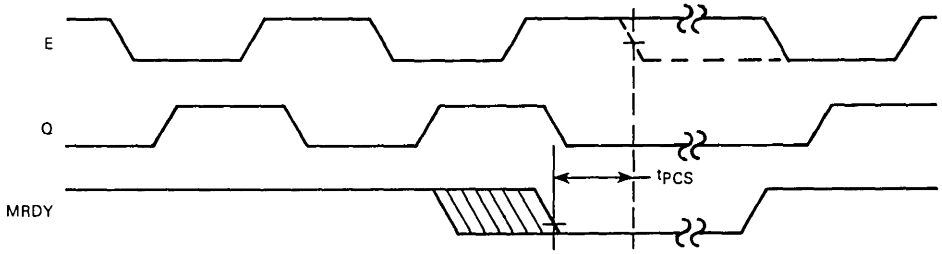
FIGURE 11 — E/Q RELATIONSHIP



NOTE· Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified

FIGURE 12 — MRDY TIMING AND SYNCHRONIZATION

(a) Timing


E
Q
MRDY
tPCS

(b) Synchronization

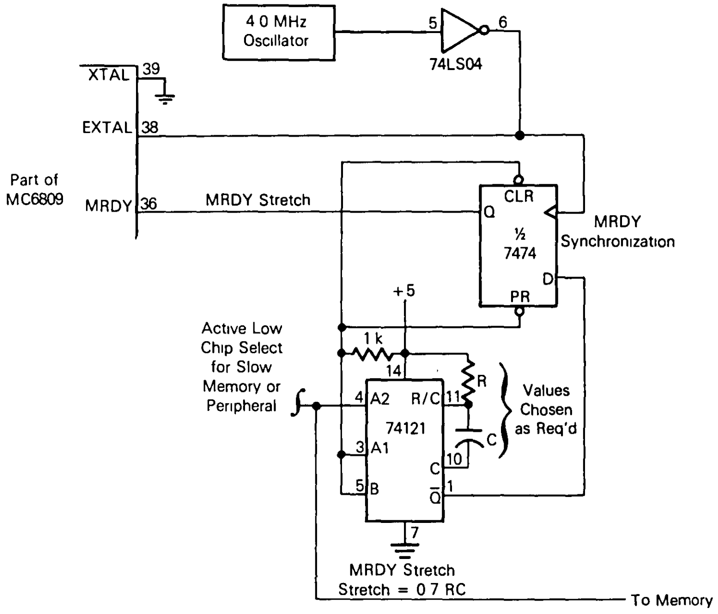

4.0 MHz Oscillator
5
6
74LS04
XTAL
39
EXTAL
38
Part of MC6809
MRDY
36
MRDY Stretch
CLR
Q
½ 7474
MRDY Synchronization
D
PR
+5
Active Low Chip Select for Slow Memory or Peripheral
1 k
14
R
4
A2
R/C
11
Values Chosen as Req'd
74121
C
3
A1
10
C
5
B
Q̄
1
7
MRDY Stretch
Stretch = 0.7 RC
To Memory

FIGURE 13 — TYPICAL DMA TIMING (<14 CYCLES)

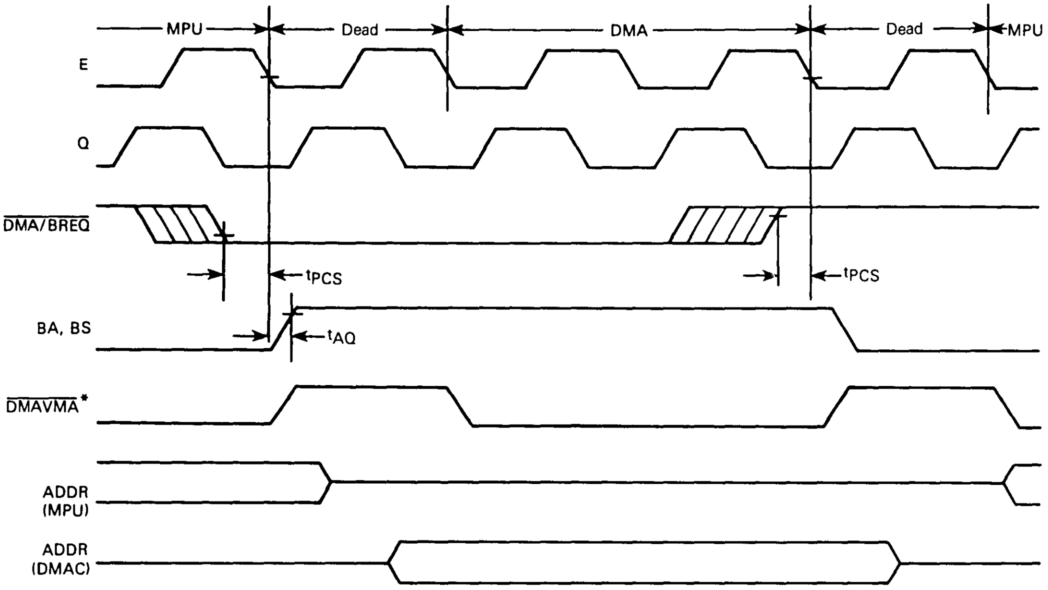
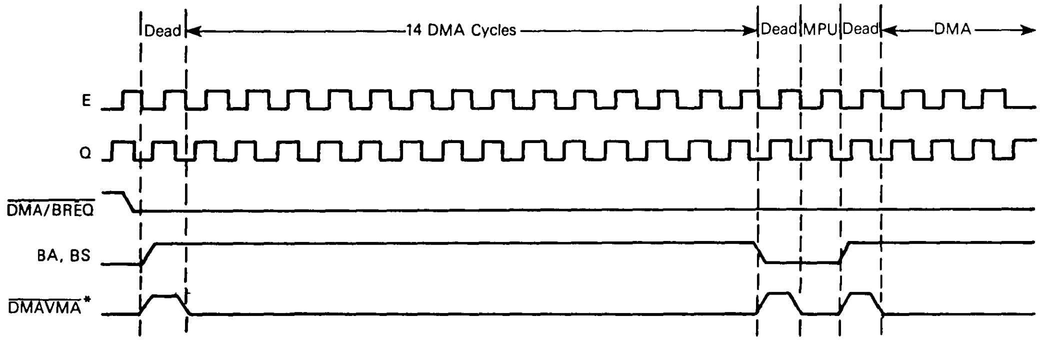

MPU
Dead
DMA
Dead
MPU
E
Q
DMA/BREQ
tPCS
tPCS
BA, BS
tAQ
DMAVMA*
ADDR (MPU)
ADDR (DMAC)

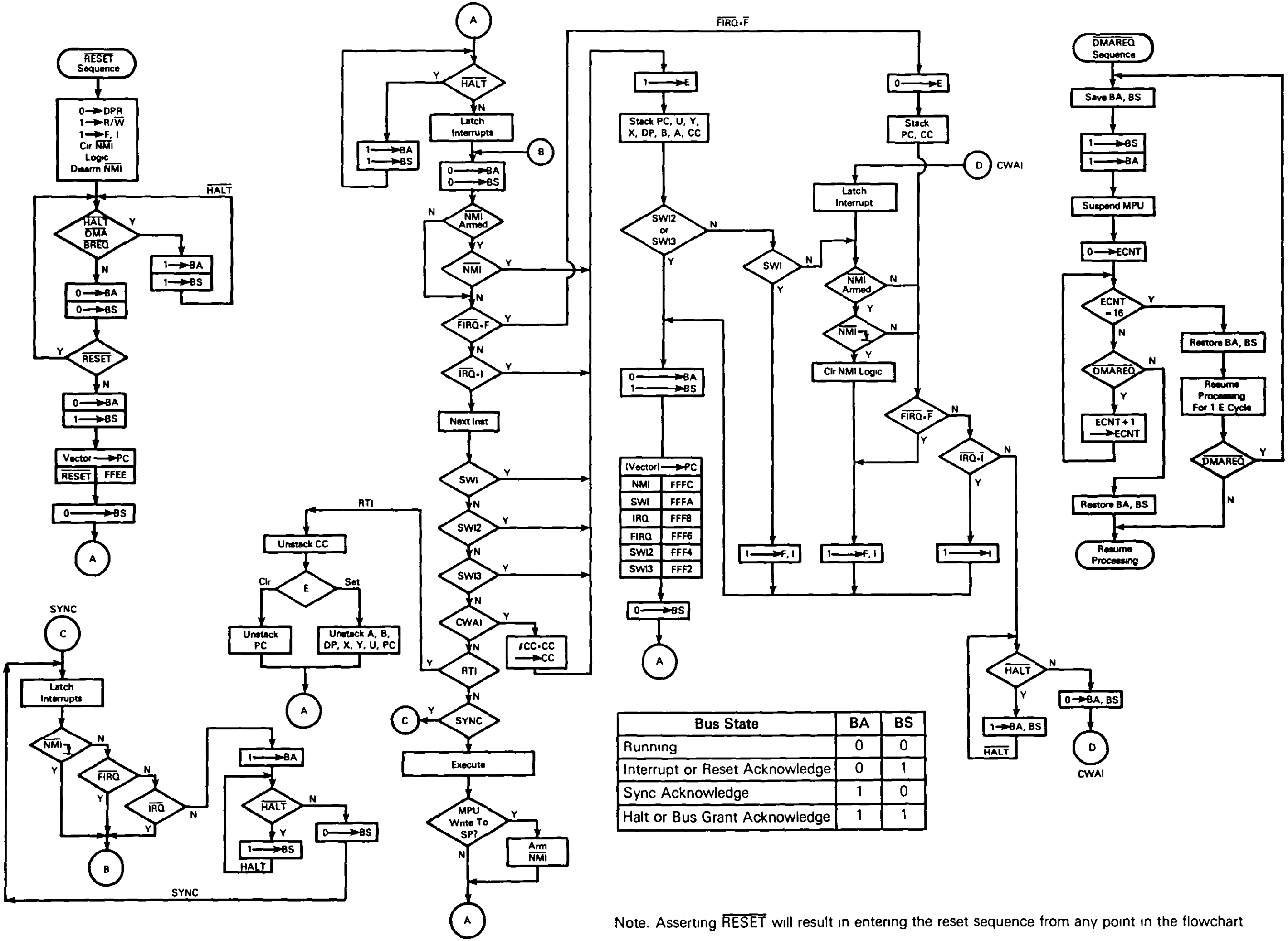
FIGURE 14 — AUTO-REFRESH DMA TIMING (>14 CYCLES)
(REVERSE CYCLE STEALING)


Dead
14 DMA Cycles
Dead
MPU
Dead
DMA
E
Q
DMA/BREQ
BA, BS
DMAVMA*

* $\overline{\text{DMAVMA}}$ is a signal which is developed externally, but is a system requirement for DMA. Refer to Application Note AN-820

NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

FIGURE 15 — FLOWCHART FOR MC6809 INSTRUCTIONS


RESET Sequence
0→DPR
1→R/W
1→F, I
Clr NMI Logic
Disarm NMI
HALT
HALT DMA BREQ
1→BA
1→BS
0→BA
0→BS
RESET
Vector→PC
RESET FFFE
0→BS
A
Y
N
HALT
Latch Interrupts
B
NMI Armed
NMI
FIRQ·F
IRQ·I
Next Inst
SWI
SWI2
SWI3
CWAI
RTI
SYNC
Execute
MPU Write To SP?
Arm NMI
#CC·CC→CC
RTI
Unstack CC
E
Clr
Set
Unstack PC
Unstack A, B, DP, X, Y, U, PC
SYNC
C
Latch Interrupts
NMI
FIRQ
IRQ
1→BA
HALT
1→BS
0→BS
SYNC
1→E
Stack PC, U, Y, X, DP, B, A, CC
SWI2 or SWI3
0→BA
1→BS
(Vector)→PC
NMI FFFC
SWI FFFA
IRQ FFF8
FIRQ FFF6
SWI2 FFF4
SWI3 FFF2
0→BS
FIRQ·F
0→E
Stack PC, CC
D
CWAI
Latch Interrupt
SWI
NMI Armed
NMI
Clr NMI Logic
FIRQ·F
IRQ·I
1→F, I
1→F, I
1→I
HALT
1→BA, BS
0→BA, BS
DMAREQ Sequence
Save BA, BS
1→BS
1→BA
Suspend MPU
0→ECNT
ECNT = 16
DMAREQ
ECNT+1→ECNT
Restore BA, BS
Resume Processing For 1 E Cycle
DMAREQ
Restore BA, BS
Resume Processing

Bus State	BA	BS
Running	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt or Bus Grant Acknowledge	1	1

Note. Asserting RESET will result in entering the reset sequence from any point in the flowchart

ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any microcomputer. For example, the MC6809 has 59 basic instructions; however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the MC6809:

- Inherent (Includes Accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary Examples of inherent addressing are: ABX, DAA, SWI, ASRA, and CLRB.

IMMEDIATE ADDRESSING

In immediate addressing, the effective address of the data is the location immediately following the opcode (i.e , the data to be used in the instruction immediately following the opcode of the instruction). The MC6809 uses both 8- and 16-bit immediate values depending on the size of argument specified by the opcode. Examples of instructions with immediate addressing are·

```
LDA #$20
LDX #$F000
LDY #CAT
```

NOTE

# signifies immediate addressing; $ signifies hexadecimal value.

EXTENDED ADDRESSING

In extended addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent Examples of extended addressing include:

```
LDA    CAT
STX    MOUSE
LDD    $2000
```

EXTENDED INDIRECT – As in the special case of indexed addressing (discussed below), one level of indirection may be added to extended addressing In extended indirect, the two bytes following the postbyte of an indexed instruction contain the address of the data

```
LDA    [CAT]
LDX    [$FFFE]
STU    [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower eight bits of the address to be used. The upper eight bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register Since the DP register is set to $00 on reset, direct addressing on the MC6809 is compatible with direct addressing on the M6800. Indirection is not allowed in direct addressing. Some examples of direct addressing are:

```
LDA     $30
SETDP   $10 (assembler directive)
LDB     $1030
LDD     < CAT
```

NOTE

< is an assembler directive which forces direct addressing

REGISTER ADDRESSING

Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction. This is called a postbyte. Some examples of register addressing are:

TFR	X, Y	Transfers X into Y
EXG	A, B	Exchanges A with B
PSHS	A, B, X, Y	Push Y, X, B and A onto S
PULU	X, Y, D	Pull D, X, and Y from U

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction Five basic types of indexing are available and are discussed below The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode as well as the pointer register to be used. Figure 16 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation.

FIGURE 16 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Postbyte Register Bit 7	6	5	4	3	2	1	0	Indexed Addressing Mode
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R+ +
1	R	R	0	0	0	1	0	,− R
1	R	R	i	0	0	1	1	,− − R
1	R	R	i	0	1	0	0	EA = ,R +0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R +8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R +16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R +D Offset
1	x	x	i	1	1	0	0	EA = ,PC +8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC +16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]

Addressing Mode Field

Indirect Field (Sign bit when b_7 = 0)

Register Field RR
00 = X
01 = Y
10 = U
11 = S

x = Don't Care
d = Offset Bit
i = 0 = Not Indirect
1 = Indirect

ZERO-OFFSET INDEXED — In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

LDD 0, X
LDA ,S

CONSTANT OFFSET INDEXED — In this mode, a two's-complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offset are available:

5-bit (−16 to +15)
8-bit (−128 to +128)
16-bit (−32768 to +32767)

The twos complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles. The twos complement 8-bit offset is contained in a single byte following the postbyte. The twos complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be cocerned with the size of this offset since the assembler will select the optimal size automatically.

Examples of constant-offset indexing are:

LDA 23,X
LDX −2,S
LDY 300,X
LDU CAT,Y

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect Assembler Form	Non Indirect Postbyte Opcode	+ ~	+ #	Indirect Assembler Form	Indirect Postbyte Opcode	+ ~	+ #
Constant Offset From R	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
(2s Complement Offsets)	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R (2s Complement Offsets)	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R+ +	1RR00001	3	0	[,R+ +]	1RR10001	6	0
	Decrement By 1	,− R	1RR00010	2	0	not allowed			
	Decrement By 2	,− − R	1RR00011	3	0	[,− − R]	1RR10011	6	0
Constant Offset From PC (2s Complement Offsets)	8-Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
	16-Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16-Bit Address	–	–	–	–	[n]	10011111	5	2

R = X, Y, U, or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

+~ and +# indicate the number of additional cycles and bytes for the particular variation

ACCUMULATOR-OFFSET INDEXED — This mode is similar to constant offset indexed except that the twos-complement value in one of the accumulators (A, B, or D) and the contents of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are:

```
LDA    B,Y
LDX    D,Y
LEAX   B,X
```

AUTO INCREMENT/DECREMENT INDEXED — In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or for the creation of software stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment; but the tables, etc., are scanned from the high to low addresses. The size of the increment/decrement can be either one or two to allow for tables of either 8- or 16-bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allows them to be used to create additional software stacks that behave identically to the U and S stacks.

Some examples of the auto increment/decrement addressing modes are:

```
LDA    ,X+
STD    ,Y++
LDB    ,-Y
LDX    ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address.

Consider the following instruction:

STX 0,X++ (X initialized to 0)

The desired result is to store zero in locations $0000 and $0001, then increment X to point to $0002. In reality, the following occurs:

0→temp	calculate the EA; temp is a holding register
X+2→X	perform auto increment
X→(temp)	do store operation

INDEXED INDIRECT — All of the indexing modes, with the exception of auto increment/decrement by one or a ±4-bit offset, may have an additional level of indirection specified. In indirect addressing, the effective address is contained at the location specified by the contents of the index register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the index register and an offset

```
        Before Execution
        A=XX (don't care)
        X=$F000
$0100    LDA  [$10,X]    EA is now $F010

$F010    $F1             $F150 is now the
$F011    $50             new EA

$F150    $AA
        After Execution
        A=$AA Actual Data Loaded
        X=$F000
```

All modes of indexed indirect are included except those which are meaningless (e.g , auto increment/decrement by one indirect) Some examples of indexed indirect are.

```
LDA    [,X]
LDD    [10,S]
LDA    [B,Y]
LDD    [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter If the branch condition is true, then the calculated address (PC + signed offset) is loaded into the program counter Program execution continues at the new location as indicated by the PC; short (one byte offset) and long (two bytes offset) relative addressing modes are available All of memory can be reached in long relative addressing as an effective address is interpreted modulo 2^{16}. Some examples of relative addressing are:

```
          BEQ     CAT       (short)
          BGT     DOG       (short)
CAT       LBEQ    RAT       (long)
DOG       LBGT    RABBIT    (long)
           •
           •
           •
RAT       NOP
RABBIT    NOP
```

PROGRAM COUNTER RELATIVE — The PC can be used as the pointer register with 8- or 16-bit signed offsets As in relative addressing, the offset is added to the current PC to create the effective address The effective address is then used as the address of the operand or data Program counter relative addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the program counter Examples are

```
LDA     CAT, PCR
LEAX    TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available

```
LDA     [CAT, PCR]
LDU     [DOG, PCR]
```

INSTRUCTION SET

The instruction set of the MC6809 is similar to that of the MC6800 and is upward compatible at the source code level. The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464.

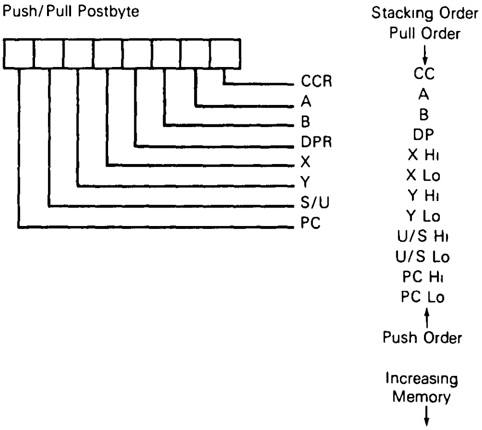
Some of the new instructions are described in detail below.

PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register or set of registers with a single instruction

PULU/PULS

The pull instructions have the same capability of the push instruction, in reverse order The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled The actual push/pull sequence is fixed, each bit defines a unique register to push or pull, as shown below



TFR/EXG

Within the MC6809, any register may be transferred to or exchanged with another or like size, i.e., 8 bit to 8 bit or 16 bit to 16 bit. Bits 4-7 of post byte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

Transfer/Exchange Postbyte

Source
Destination

Register Field

0000 = D (A:B)	1000 = A
0001 = X	1001 = B
0010 = Y	1010 = CCR
0011 = U	1011 = DPR
0100 = S	
0101 = PC	

NOTE

All other combinations are undefined and INVALID

LEAX/LEAY/LEAU/LEAS

The LEA (load effective address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register This makes all the features of the internal addressing hardware available to the programmer. Some of the implications of this instruction are illustrated in Table 3

The LEA instruction also allows the user to access data and tables in a position independent manner For example:

```
        LEAX    MSG1, PCR
        LBSR    PDATA (print message routine)
        •
        •
MSG1    FCC     'MESSAGE'
```

This sample program prints: 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1 This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the auto increment and auto decrement addressing modes due to the sequence of internal operations. The LEA internal sequence is outlined as follows.

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U, or S may be substituted for a and b)

1. b → temp	(calculate the EA)
2. b + 1 → b	(modify b, postincrement)
3 temp → a	(load a)

LEAa , – b

1. b – 1 → temp	(calculate EA with predecrement)
2. b – 1 → b	(modify b, predecrement)
3. temp → a	(load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-Bit Constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-Bit Constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-Bit A Accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-Bit D Accumulator to Y
LEAU – 10, U	U – 10 → U	Substracts 10 from U
LEAS – 10, S	S – 10 → S	Used to Reserve Area on Stack
LEAS 10, S	S + 10 → S	Used to 'Clean Up' Stack
LEAX 5, S	S + 5 → X	Transfers As Well As Adds

Auto increment-by-two and auto decrement-by-two instructions work similarly. Note that LEAX ,X+ does not change X; however, LEAX, −X does decrement; LEAX 1, X should be used to increment X by one.

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator. The unsigned multiply also allows multiple-precision multiplications.

LONG AND SHORT RELATIVE BRANCHES

The MC6809 has the capability of program counter relative branching throughout the entire memory map. In this mode, if the branch is to be taken, the 8- or 16-bit signed offset is added to the value of the program counter to be used as the effective address. This allows the program to branch anywhere in the 64K memory map. Position-independent code can be easily generated through the use of relative branching. Both short (8-bit) and long (16-bit) branches are available.

SYNC

After encountering a sync instruction, the MPU enters a sync state, stops processing instructions, and waits for an interrupt If the pending interrupt is non-maskable ($\overline{\text{NMI}}$) or maskable ($\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$) with its mask bit (F or I) clear, the processor will clear the sync state and perform the normal interrupt stacking and service routine. Since $\overline{\text{FIRQ}}$ and $\overline{\text{IRQ}}$ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken. If the pending interrupt is maskable ($\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$) with its mask bit (F or I) set, the processor will clear the sync state and continue processing by executing the next in-line instruction. Figure 18 depicts sync timing.

SOFTWARE INTERRUPTS

A software interrupt is an instruction which will cause an interrupt and its associated vector fetch. These software interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems. Three levels of SWI are available on the MC6809, and are prioritized in the following order: SWI, SWI2, SWI3.

16-BIT OPERATION

The MC6809 has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes, and pulls

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart (Figure 18) illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the MC6809. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flowchart. $\overline{\text{VMA}}$ is an indication of $FFFF_{16}$ on the address bus, $R/\overline{W} = 1$ and $BS = 0$. The following examples illustrate the use of the chart.

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                    •
                    •
                    •
$8000               LBSR   CAT
                    •
                    •
                    •
$A000     CAT       •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	$R/\overline{W}$	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
5	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000     DEC      $A000
          •
          •
          •
$A8000    $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	$R/\overline{W}$	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
7	A000	7F	0	Store the Decremented Data

*The data bus has the data at that particular address

INSTRUCTION SET TABLES

The instructions of the MC6809 have been broken down into five different categories. They are as follows:

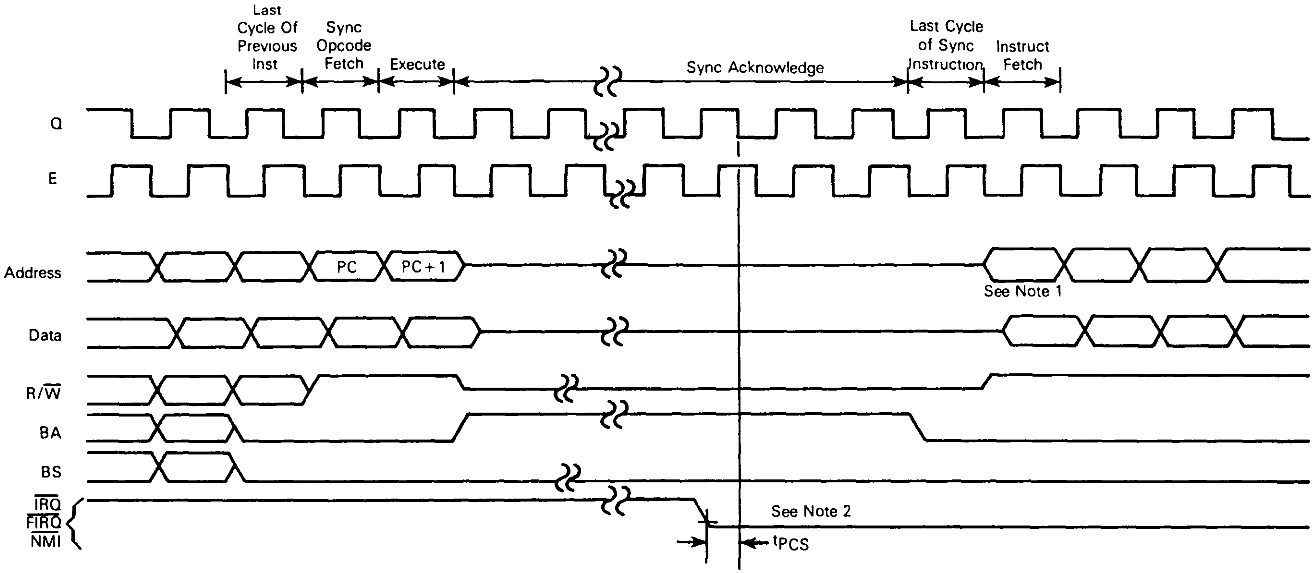
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 19 contains a compilation of data that will assist in programming the MC6809.

FIGURE 17 — SYNC TIMING


Last Cycle Of Previous Inst
Sync Opcode Fetch
Execute
Sync Acknowledge
Last Cycle of Sync Instruction
Instruct Fetch
Q
E
Address
PC
PC+1
See Note 1
Data
R/$\overline{W}$
BA
BS
$\overline{IRQ}$
$\overline{FIRQ}$
$\overline{NMI}$
See Note 2
t_{PCS}

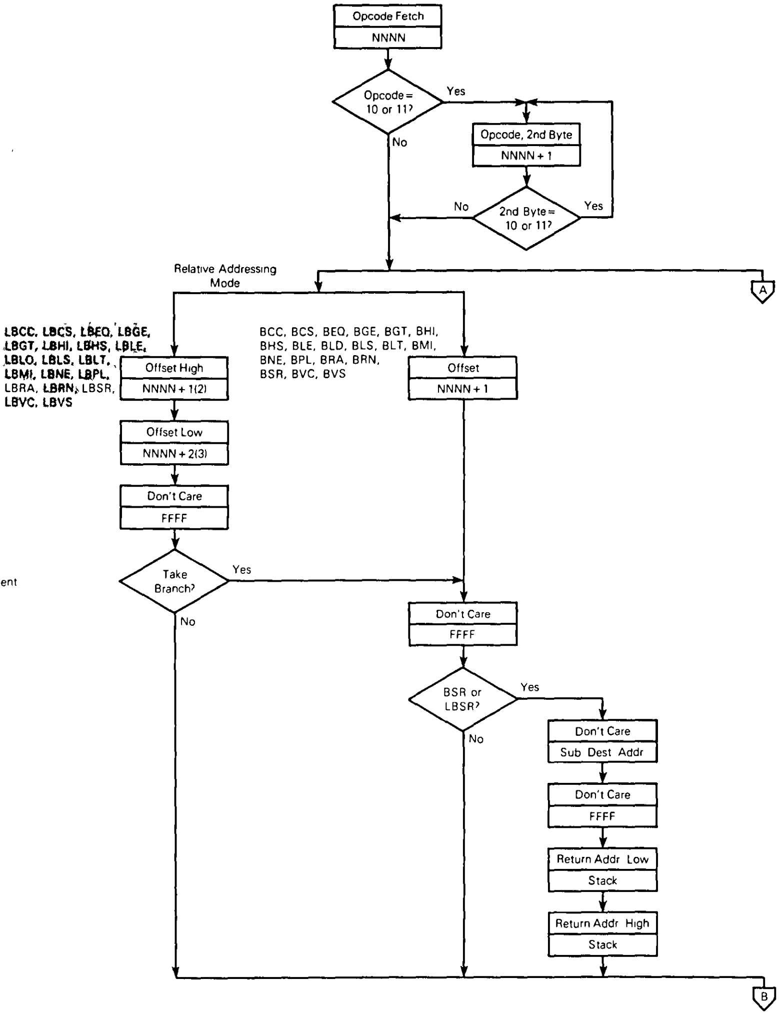
NOTES

1 If the associated mask bit is set when the interrupt is requested, this cycle will be an instruction fetch from address location PC+1 However, if the interrupt is accepted ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) interrupt processing continues with this cycle as in on Figures 9 and 10 (Interrupt Timing)

2 If mask bits are clear, $\overline{IRQ}$ and $\overline{FIRQ}$ must be held low for three cycles to guarantee interrupt to be taken, although only one cycle is necessary to bring the processor out of SYNC

3 Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 5)


Opcode Fetch
NNNN
Opcode = 10 or 11?
Yes
No
Opcode, 2nd Byte
NNNN + 1
2nd Byte = 10 or 11?
No
Yes
Relative Addressing Mode
A
LBCC, LBCS, LBEQ, LBGE, LBGT, LBHI, LBHS, LBLE, LBLO, LBLS, LBLT, LBMI, LBNE, LBPL, LBRA, LBRN, LBSR, LBVC, LBVS
Offset High
NNNN + 1(2)
Offset Low
NNNN + 2(3)
Don't Care
FFFF
Take Branch?
Yes
No
BCC, BCS, BEQ, BGE, BGT, BHI, BHS, BLE, BLD, BLS, BLT, BMI, BNE, BPL, BRA, BRN, BSR, BVC, BVS
Offset
NNNN + 1
Don't Care
FFFF
BSR or LBSR?
Yes
No
Don't Care
Sub Dest Addr
Don't Care
FFFF
Return Addr Low
Stack
Return Addr High
Stack
B

NOTES
1 Each state shows

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2 Address NNNN is location of opcode

3 If opcode is a two byte opcode subsequent addresses are in parenthesis (–)

4 Two-byte opcodes are highlighted

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 5)

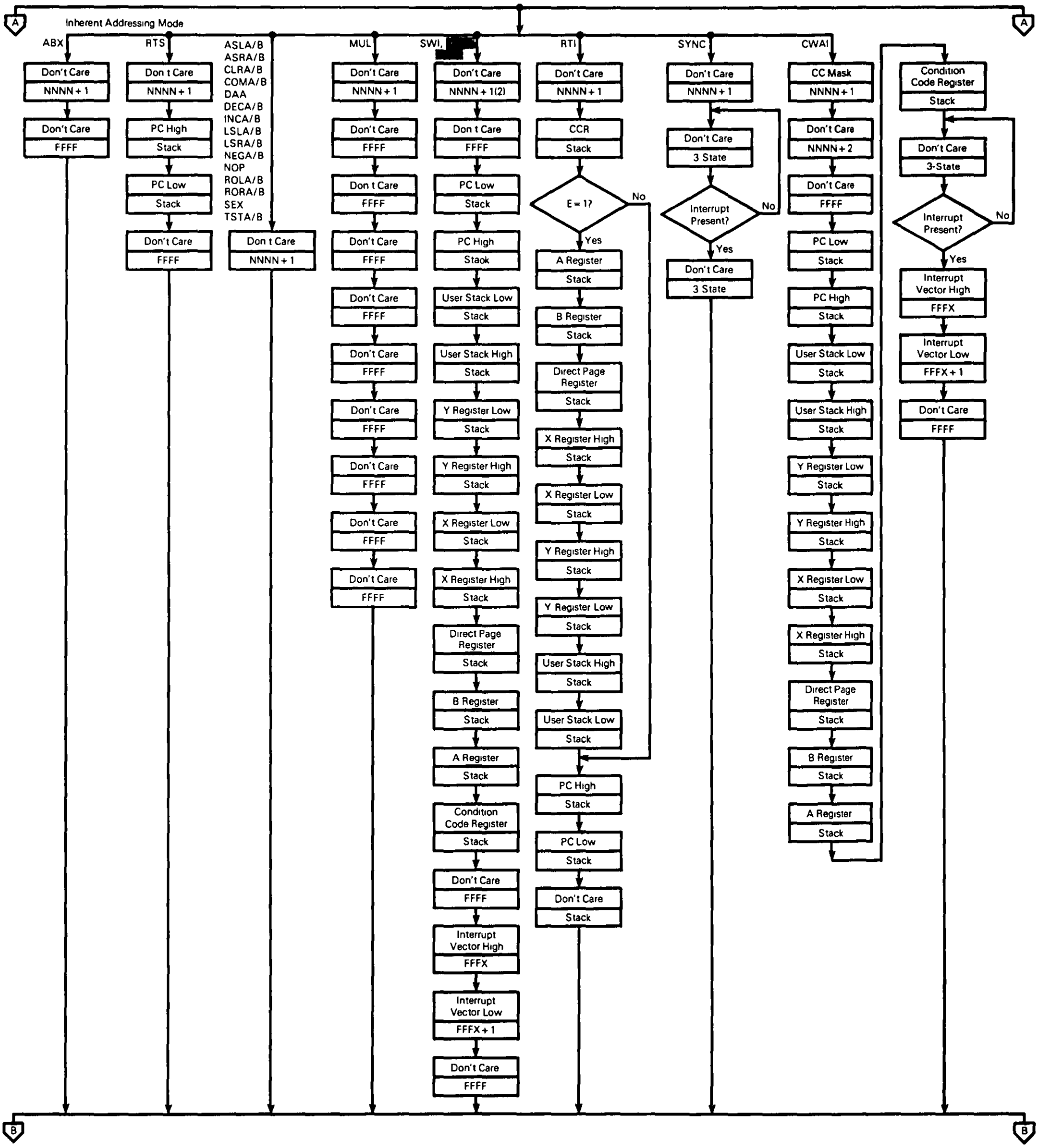

A
Inherent Addressing Mode
ABX
Don't Care
NNNN+1
Don't Care
FFFF
RTS
Don t Care
NNNN+1
PC High
Stack
PC Low
Stack
Don't Care
FFFF
ASLA/B
ASRA/B
CLRA/B
COMA/B
DAA
DECA/B
INCA/B
LSLA/B
LSRA/B
NEGA/B
NOP
ROLA/B
RORA/B
SEX
TSTA/B
Don t Care
NNNN+1
MUL
Don't Care
NNNN+1
Don't Care
FFFF
Don t Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
SWI, [redacted]
Don't Care
NNNN+1(2)
Don t Care
FFFF
PC Low
Stack
PC High
Stack
User Stack Low
Stack
User Stack High
Stack
Y Register Low
Stack
Y Register High
Stack
X Register Low
Stack
X Register High
Stack
Direct Page Register
Stack
B Register
Stack
A Register
Stack
Condition Code Register
Stack
Don't Care
FFFF
Interrupt Vector High
FFFX
Interrupt Vector Low
FFFX+1
Don't Care
FFFF
RTI
Don't Care
NNNN+1
CCR
Stack
E=1?
No
Yes
A Register
Stack
B Register
Stack
Direct Page Register
Stack
X Register High
Stack
X Register Low
Stack
Y Register High
Stack
Y Register Low
Stack
User Stack High
Stack
User Stack Low
Stack
PC High
Stack
PC Low
Stack
Don't Care
Stack
SYNC
Don't Care
NNNN+1
Don't Care
3 State
Interrupt Present?
No
Yes
Don't Care
3 State
CWAI
CC Mask
NNNN+1
Don't Care
NNNN+2
Don't Care
FFFF
PC Low
Stack
PC High
Stack
User Stack Low
Stack
User Stack High
Stack
Y Register Low
Stack
Y Register High
Stack
X Register Low
Stack
X Register High
Stack
Direct Page Register
Stack
B Register
Stack
A Register
Stack
Condition Code Register
Stack
Don't Care
3-State
Interrupt Present?
No
Yes
Interrupt Vector High
FFFX
Interrupt Vector Low
FFFX+1
Don't Care
FFFF
A
B
B

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 3 of 5)

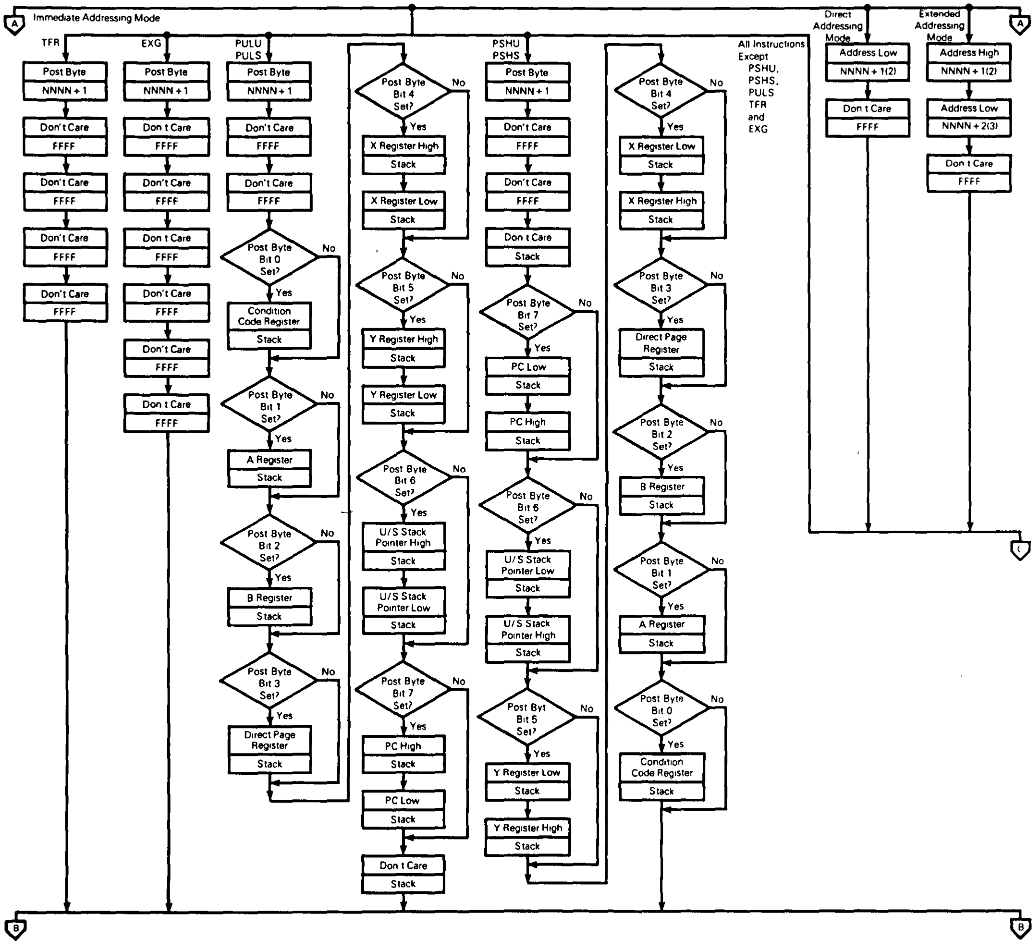
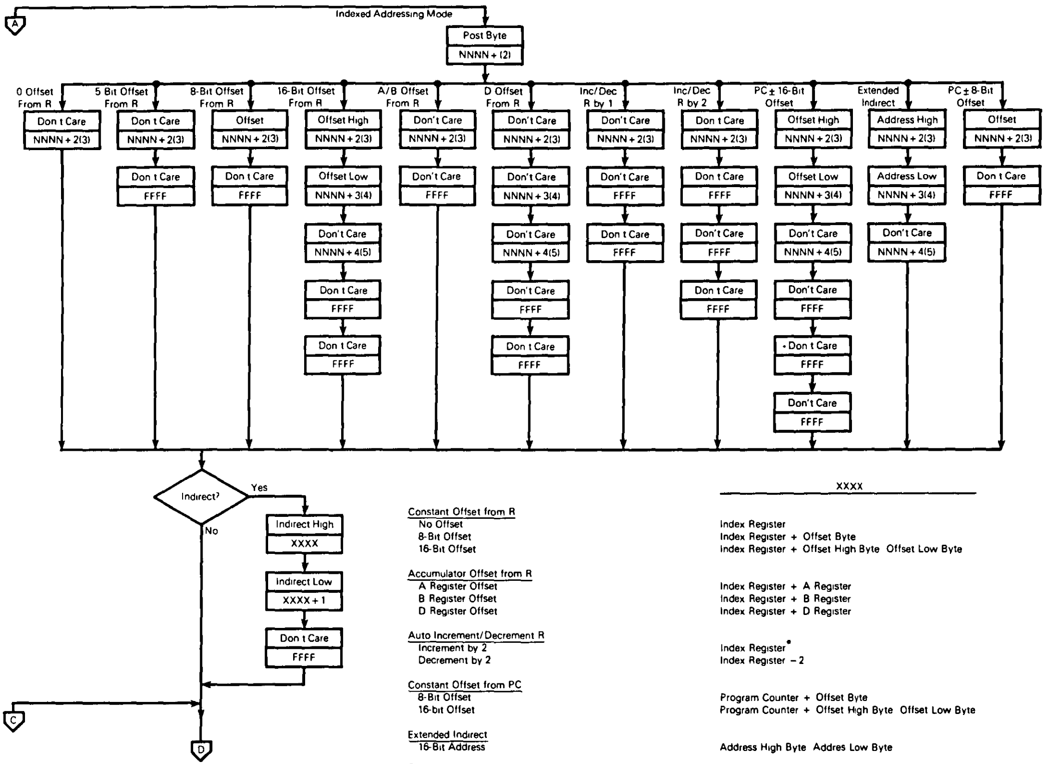

A
Immediate Addressing Mode
TFR
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
EXG
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
PULU
PULS
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Post Byte Bit 0 Set?
Yes
No
Condition Code Register
Stack
Post Byte Bit 1 Set?
Yes
No
A Register
Stack
Post Byte Bit 2 Set?
Yes
No
B Register
Stack
Post Byte Bit 3 Set?
Yes
No
Direct Page Register
Stack
Post Byte Bit 4 Set?
Yes
No
X Register High
Stack
X Register Low
Stack
Post Byte Bit 5 Set?
Yes
No
Y Register High
Stack
Y Register Low
Stack
Post Byte Bit 6 Set?
Yes
No
U/S Stack Pointer High
Stack
U/S Stack Pointer Low
Stack
Post Byte Bit 7 Set?
Yes
No
PC High
Stack
PC Low
Stack
Don't Care
Stack
PSHU
PSHS
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
Stack
Post Byte Bit 7 Set?
Yes
No
PC Low
Stack
PC High
Stack
Post Byte Bit 6 Set?
Yes
No
U/S Stack Pointer Low
Stack
U/S Stack Pointer High
Stack
Post Byt Bit 5 Set?
Yes
No
Y Register Low
Stack
Y Register High
Stack
Post Byte Bit 4 Set?
Yes
No
X Register Low
Stack
X Register High
Stack
Post Byte Bit 3 Set?
Yes
No
Direct Page Register
Stack
Post Byte Bit 2 Set?
Yes
No
B Register
Stack
Post Byte Bit 1 Set?
Yes
No
A Register
Stack
Post Byte Bit 0 Set?
Yes
No
Condition Code Register
Stack
All Instructions Except PSHU, PSHS, PULS TFR and EXG
Direct Addressing Mode
Address Low
NNNN + 1(2)
Don't Care
FFFF
Extended Addressing Mode
Address High
NNNN + 1(2)
Address Low
NNNN + 2(3)
Don't Care
FFFF
A
C
B
B

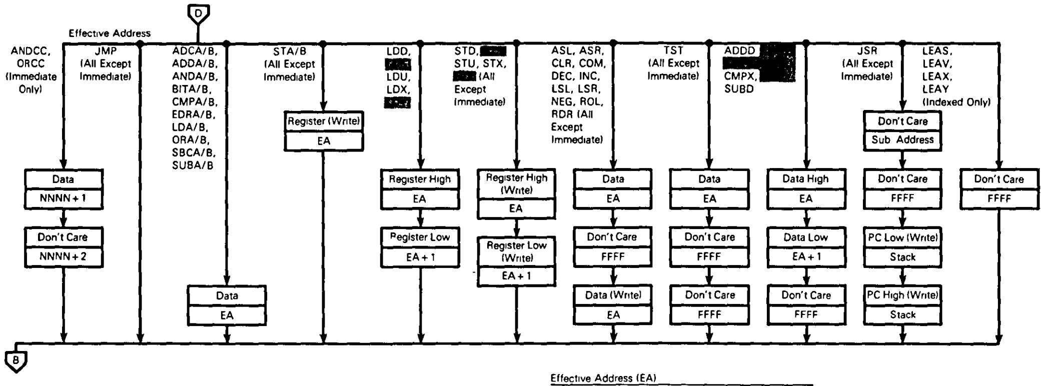
FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 5)


A
Indexed Addressing Mode
Post Byte
NNNN + (2)
0 Offset From R
5 Bit Offset From R
8-Bit Offset From R
16-Bit Offset From R
A/B Offset From R
D Offset From R
Inc/Dec R by 1
Inc/Dec R by 2
PC ± 16-Bit Offset
Extended Indirect
PC ± 8-Bit Offset
Don't Care
NNNN + 2(3)
Offset
Offset High
Offset Low
NNNN + 3(4)
NNNN + 4(5)
FFFF
Address High
Address Low
Indirect?
Yes
No
Indirect High
XXXX
Indirect Low
XXXX + 1
C
D

	XXXX
Constant Offset from R	
No Offset	Index Register
8-Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register − 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-bit Offset	Program Counter + Offset High Byte Offset Low Byte
Extended Indirect	
16-Bit Address	Address High Byte Addres Low Byte

* The index register is incremented following the indexed access

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 5)


D
Effective Address
ANDCC, ORCC (Immediate Only)
JMP (All Except Immediate)
ADCA/B, ADDA/B, ANDA/B, BITA/B, CMPA/B, EDRA/B, LDA/B, ORA/B, SBCA/B, SUBA/B
STA/B (All Except Immediate)
LDD, [redacted] LDU, LDX, [redacted]
STD, [redacted] STU, STX, [redacted] (All Except Immediate)
ASL, ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, RDR (All Except Immediate)
TST (All Except Immediate)
ADDD [redacted] CMPX, SUBD
JSR (All Except Immediate)
LEAS, LEAV, LEAX, LEAY (Indexed Only)
Data
NNNN + 1
Don't Care
NNNN + 2
Data
EA
Register (Write)
EA
Register High
EA
Register Low
EA + 1
Register High (Write)
EA
Register Low (Write)
EA + 1
Data
EA
Don't Care
FFFF
Data (Write)
EA
Data
EA
Don't Care
FFFF
Don't Care
FFFF
Data High
EA
Data Low
EA + 1
Don't Care
FFFF
Don't Care
Sub Address
Don't Care
FFFF
PC Low (Write)
Stack
PC High (Write)
Stack
Don't Care
FFFF
B

	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register − 1
Decrement by 2	Index Register − 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16 Bit Offset	Program Counter + Offset High Byte Offset Low Byte
Direct	Direct Page Register Address Low
Extended	Address High Address Low
Immediate	NNNN + 1

*The index register is incremented following the indexed access

TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE A, B, CC, or DP may be pushed to (pulled from) stack with either PSHS, PSHU (PULS, PULU) instructions

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U, or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U, or PC
TFR R, D	Transfer X, Y, S, U, or PC to D

NOTE D may be pushed (pulled) to stack with either PSHS, PSHU (PULS, PULU) instructions

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, X, U, or PC with D, X Y, S, U, or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U, or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S, or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U, or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 — BRANCH INSTRUCTIONS

Instruction	Description
	SIMPLE BRANCHES
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
	SIGNED BRANCHES
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2s complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2s complement result
BLT, LBLT	Branch if less than (signed)
	UNSIGNED BRANCHES
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
	OTHER BRANCHES
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 — MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	*	↑			31	LEAY	↑	4+	2+	61	*	↑		
02	*				32	LEAS	↓	4+	2+	62	*			
03	COM		6	2	33	LEAU	Indexed	4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*	–			68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX	↑	3	1	6A	DEC		6+	2+
0B	*				3B	RTI		6/15	1	6B	*			
0C	INC		6	2	3C	CWAI	↓	≥20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL	Inherent	11	1	6D	TST		6+	2+
0E	JMP	↓	3	2	3E	*	–			6E	JMP	↓	3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	–	–	–	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	–	–	–	41	*	↑			71	*	↑		
12	NOP	Inherent	2	1	42	*				72	*			
13	SYNC	Inherent	≥4	1	43	COMA		2	1	73	COM		7	3
14	*				44	LSRA		2	1	74	LSR		7	3
15	*				45	*				75	*			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	*				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	*	–			4B	*				7B	*			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG	Immed	8	2	4E	*	↓			7E	JMP	↓	4	3
1F	TFR	Immed	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN	↑	3	2	51	*	↑			81	CMPA	↑	2	2
22	BHI		3	2	52	*				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	*				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	*			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	*				8B	ADDA	↓	2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT	↓	3	2	5E	*	↓			8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	*			

LEGEND

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)
# Number of program bytes
* Denotes unused opcode

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

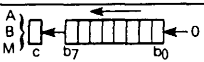
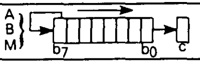
OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

NOTE. All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDU	Immed	3	3
CF	*			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

OP	Mnem	Mode	~	#
Page 2 and 3 Machine Codes				
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

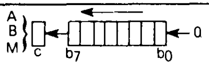
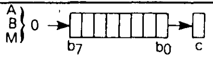
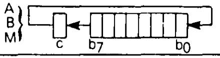
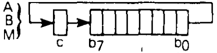
FIGURE 19 — PROGRAMMING AID

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0
		Immediate			Direct			Indexed			Extended			Inherent								
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	N	Z	V	C
ABX														3A	3	1	B + X → X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4 +	2 +	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4 +	2 +	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4 +	2 +	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4 +	2 +	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6 +	2 +	F3	7	3				D + M M + 1 → D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4 +	2 +	B4	5	3				A ∧ M → A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4 +	2 +	F4	5	3				B ∧ M → B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC ∧ IMM → CC					7
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6 +	2 +	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	•	↕
	ASRB													57	2	1		8	↕	↕	•	↕
	ASR				07	6	2	67	6 +	2 +	77	7	3					8	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4 +	2 +	B5	5	3				Bit Test A (M ∧ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4 +	2 +	F5	5	3				Bit Test B (M ∧ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0 → A	•	0	1	0	0
	CLRB													5F	2	1	0 → B	•	0	1	0	0
	CLR				0F	6	2	6F	6 +	2 +	7F	7	3				0 → M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4 +	2 +	B1	5	3				Compare M from A	8	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4 +	2 +	F1	5	3				Compare M from B	8	↕	↕	↕	↕
	CMPD	10 83	5	4	10 93	7	3	10 A3	7 +	3 +	10 B3	8	4				Compare M M + 1 from D	•	↕	↕	↕	↕
	CMPS	11 8C	5	4	11 9C	7	3	11 AC	7 +	3 +	11 BC	8	4				Compare M M + 1 from S	•	↕	↕	↕	↕
	CMPU	11 83	5	4	11 93	7	3	11 A3	7 +	3 +	11 B3	8	4				Compare M M + 1 from U	•	↕	↕	↕	↕
	CMPX	8C	4	3	9C	6	2	AC	6 +	2 +	BC	7	3				Compare M M + 1 from X	•	↕	↕	↕	↕
	CMPY	10 8C	5	4	10 9C	7	3	10 AC	7 +	3 +	10 BC	8	4				Compare M M + 1 from Y	•	↕	↕	↕	↕
COM	COMA													43	2	1	$\overline{A}$ → A	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B}$ → B	•	↕	↕	0	1
	COM				03	6	2	63	6 +	2 +	73	7	3				$\overline{M}$ → M	•	↕	↕	0	1
CWAI		3C	≥20	2													CC ∧ IMM → CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6 +	2 +	7A	7	3				M − 1 → M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4 +	2 +	B8	5	3				A ⊻ M → A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4 +	2 +	F8	5	3				B ⊻ M → B	•	↕	↕	0	•
EXG	R1, R2	1E	8	2													R1 ↔ R2^2	•	•	•	•	•
INC	INCA													4C	2	1	A + 1 → A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6 +	2 +	7C	7	3				M + 1 → M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3 +	2 +	7E	4	3				EA^3 → PC	•	•	•	•	•
JSR					9D	7	2	AD	7 +	2 +	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4 +	2 +	B6	5	3				M → A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4 +	2 +	F6	5	3				M → B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5 +	2 +	FC	6	3				M M + 1 → D	•	↕	↕	0	•
	LDS	10 CE	4	4	10 DE	6	3	10 EE	6 +	3 +	10 FE	7	4				M M + 1 → S	•	↕	↕	0	•
	LDU	CE	3	3	DE	5	2	EE	5 +	2 +	FE	6	3				M M + 1 → U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5 +	2 +	BE	6	3				M M + 1 → X	•	↕	↕	0	•
	LDY	10 8E	4	4	10 9E	6	3	10 AE	6 +	3 +	10 BE	7	4				M M + 1 → Y	•	↕	↕	0	•
LEA	LEAS							32	4 +	2 +							EA^3 → S	•	•	•	•	•
	LEAU							33	4 +	2 +							EA^3 → U	•	•	•	•	•
	LEAX							30	4 +	2 +							EA^3 → X	•	•	↕	•	•
	LEAY							31	4 +	2 +							EA^3 → Y	•	•	↕	•	•

LEGEND

OP	Operation Code (Hexadecimal)	$\overline{M}$	Complement of M	↕	Test and set if true, cleared otherwise
~	Number of MPU Cycles	→	Transfer Into	•	Not Affected
#	Number of Program Bytes	H	Half-carry (from bit 3)	CC	Condition Code Register
+	Arithmetic Plus	N	Negative (sign bit)		Concatenation
−	Arithmetic Minus	Z	Zero result	∨	Logical or
•	Multiply	V	Overflow, 2's complement	∧	Logical and
		C	Carry from ALU	⊻	Logical Exclusive or

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0
		Immediate			Direct			Indexed[1]			Extended			Inherent				H	N	Z	V	C
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#						
LSL	LSLA													48	2	1		•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1		•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}+1 \to A$	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}+1 \to B$	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}+1 \to M$	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A V M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B V M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC V IMM → CC				7	
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1		•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1		•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A − M − C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B − M − C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M M + 1	•	↕	↕	0	•
	STS				10	6	3	10	6+	3+	10	7	4				S → M M + 1	•	↕	↕	0	•
					DF			EF			FF											
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M M + 1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M M + 1	•	↕	↕	0	•
	STY				10	6	3	10			10	7	4				Y → M M + 1	•	↕	↕	0	•
					9F			AF	6+	3+	BF											
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A − M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B − M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D − M M + 1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI2[6]													10	20	2	Software Interrupt 2	•	•	•	•	•
														3F								
	SWI3[6]													11	20	1	Software Interrupt 3	•	•	•	•	•
														3F								
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2	1F	6	2													R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

NOTES:

1. This column gives a base cycle and byte count To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions)
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Branch Instructions

		Addressing Mode				5	3	2	1	0
		Relative								
Instruction	Forms	OP	~5	#	Description	H	N	Z	V	C
BCC	BCC	24	3	2	Branch C=0	•	•	•	•	•
	LBCC	10 24	5(6)	4	Long Branch C=0	•	•	•	•	•
BCS	BCS	25	3	2	Branch C=1	•	•	•	•	•
	LBCS	10 25	5(6)	4	Long Branch C=1	•	•	•	•	•
BEQ	BEQ	27	3	2	Branch Z=1	•	•	•	•	•
	LBEQ	10 27	5(6)	4	Long Branch Z=0	•	•	•	•	•
BGE	BGE	2C	3	2	Branch≥Zero	•	•	•	•	•
	LBGE	10 2C	5(6)	4	Long Branch≥Zero	•	•	•	•	•
BGT	BGT	2E	3	2	Branch>Zero	•	•	•	•	•
	LBGT	10 2E	5(6)	4	Long Branch>Zero	•	•	•	•	•
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10 22	5(6)	4	Long Branch Higher	•	•	•	•	•
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10 24	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
BLE	BLE	2F	3	2	Branch≤Zero	•	•	•	•	•
	LBLE	10 2F	5(6)	4	Long Branch≤Zero	•	•	•	•	•
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10 25	5(6)	4	Long Branch Lower	•	•	•	•	•

		Addressing Mode				5	3	2	1	0
		Relative								
Instruction	Forms	OP	~5	#	Description	H	N	Z	V	C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10 23	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
BLT	BLT	2D	3	2	Branch<Zero	•	•	•	•	•
	LBLT	10 2D	5(6)	4	Long Branch<Zero	•	•	•	•	•
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10 2B	5(6)	4	Long Branch Minus	•	•	•	•	•
BNE	BNE	26	3	2	Branch Z=0	•	•	•	•	•
	LBNE	10 26	5(6)	4	Long Branch Z≠0	•	•	•	•	•
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10 2A	5(6)	4	Long Branch Plus	•	•	•	•	•
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10 21	5	4	Long Branch Never	•	•	•	•	•
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V=0	•	•	•	•	•
	LBVC	10 28	5(6)	4	Long Branch V=0	•	•	•	•	•
BVS	BVS	29	3	2	Branch V=1	•	•	•	•	•
	LBVS	10 29	5(6)	4	Long Branch V=1	•	•	•	•	•

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N=1	BMI	2B	BPL	2A
Z=1	BEQ	27	BNE	26
V=1	BVS	29	BVC	28
C=1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r>m	BGT	2E	BLE	2F
r≥m	BGE	2C	BLT	2D
r=m	BEQ	27	BNE	26
r≤m	BLE	2F	BGT	2E
r<m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r>m	BHI	22	BLS	23
r≥m	BHS	24	BLO	25
r=m	BEQ	27	BNE	26
r≤m	BLS	23	BHI	22
r<m	BLO	25	BHS	24

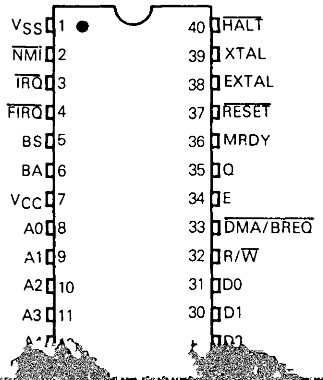
NOTES

1 All conditional branches have both short and long variations
2 All short branches are two bytes and require three cycles
3 All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4 All conditional long branches require four bytes and six cycles if the branch is taken or five cycles if the branch is not taken.

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6809P
	1.0 MHz	−40°C to 85°C	MC6809CP
	1.5 MHz	0°C to 70°C	MC68A09P
	1.5 MHz	−40°C to 85°C	MC68A09CP
	2.0 MHz	0°C to 70°C	MC68B09P
	2.0 MHz	−40°C to 85°C	MC68B09CP
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6809S
	1.0 MHz	−40°C to 85°C	MC6809CS
	1.5 MHz	0°C to 70°C	MC68A09S
	1.5 MHz	−40°C to 85°C	MC68A09CS
	2.0 MHz	0°C to 70°C	MC68B09S
	2.0 MHz	−40°C to 85°C	MC68B09CS

PIN ASSIGNMENT


V_{SS} 1
$\overline{NMI}$ 2
$\overline{IRQ}$ 3
$\overline{FIRQ}$ 4
BS 5
BA 6
V_{CC} 7
A0 8
A1 9
A2 10
A3 11
40 $\overline{HALT}$
39 XTAL
38 EXTAL
37 $\overline{RESET}$
36 MRDY
35 Q
34 E
33 $\overline{DMA/BREQ}$
32 R/$\overline{W}$
31 D0
30 D1

MC6809E

8-Bit Microprocessing Unit

The MC6809E is a high-performance 8-bit microprocessor which supports modern programming techiques such as position independence, re-entrancy, and modular programming.

This third-generation addition to the M6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809E has the most complete set of addressing modes available on any 8-bit microprocessor.

The MC6809E has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications. External clock inputs are provided to allow synchronization with peripherals, systems, or other MPUs.

MC6800 COMPATIBLE

- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- External Clock Inputs, E and Q, Allow Synchronization
- TSC Input Controls Internal Bus Buffers
- LIC Indicates Opcode Fetch
- AVMA Allows Efficient Use of Common Resources in a Multiprocessor System
- BUSY is a Status Line for Multiprocessing
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- Interrupt Acknowledge Output Allows Vectoring By Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use With Slower Memories
- Early Write Data for Dynamic Memories

SOFTWARE FEATURES

- 10 Addressing Modes
  - M6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing
    0-, 5-, 8-, or 16-Bit Constant Offsets
    8- or 16-Bit Accumulator Offsets
    Auto-Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instruction with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-Bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6809E, MC68A09E, MC68B09E MC6809EC, MC68A09EC, MC68B09EC	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Cerdip		60	
Plastic		100	

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \tag{1}$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{PORT}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \tag{2}$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \tag{3}$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

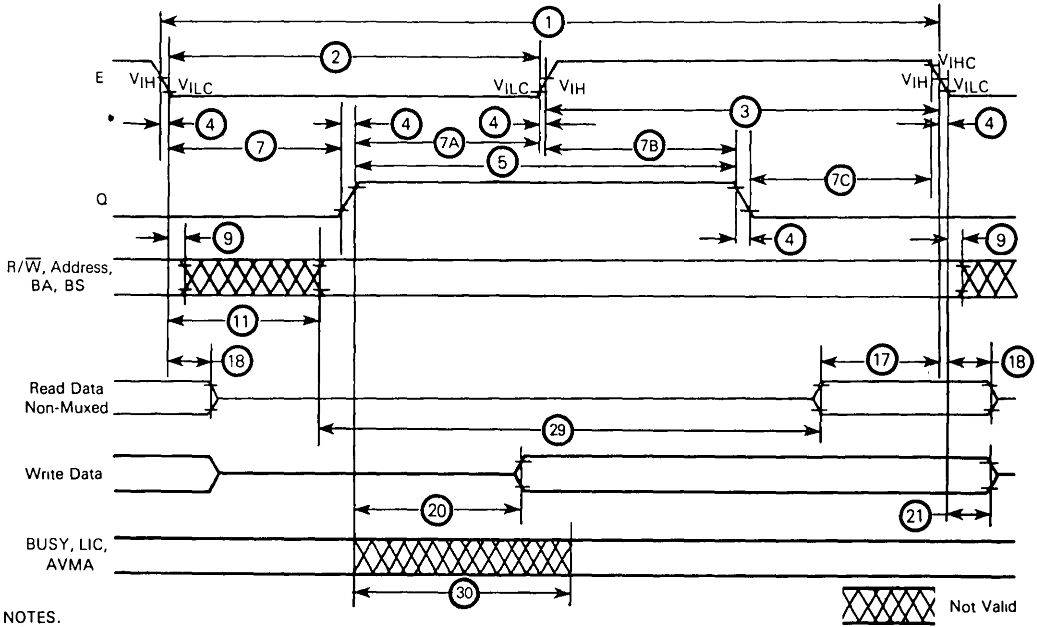
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, Q,	V_{IH}	V_{SS} + 2.0		V_{CC}	V
	$\overline{\text{RESET}}$	V_{IHR}	V_{SS} + 4.0	–	V_{CC}	
	E	V_{IHC}	V_{CC} − 0.75	–	V_{CC} + 0.3	
Input Low Voltage	Logic, $\overline{\text{RESET}}$	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
	E	V_{ILC}	V_{SS} − 0.3	–	V_{SS} + 0.4	V
	Q	V_{ILQ}	V_{SS} − 0.3	–	V_{SS} + 0.6	V
Input Leakage Current	Logic, Q, $\overline{\text{RESET}}$	I_{in}	–	–	2.5	μA
(V_{in} = 0 to 5.25 V, V_{CC} = max)	E		–	–	100	
dc Output High Voltage		V_{OH}				V
(I_{Load} = −205 μA, V_{CC} = min)	D0-D7		V_{SS} + 2.4	–	–	
(I_{Load} = −145 μA, V_{CC} = min)	A0-A15, R/$\overline{\text{W}}$		V_{SS} + 2.4	–	–	
(I_{Load} = −100 μA, V_{CC} = min)	BA, BS, LIC, AVMA, BUSY		V_{SS} + 2.4	–	–	
dc Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS} + 0.5	V
Internal Power Dissipation (Measured at T_A = 0°C in Steady State Operation)		P_{INT}	–	–	1.0	W
Capacitance		C_{in}				
(V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, Logic Inputs, Q, $\overline{\text{RESET}}$		–	10	15	pF
	E		–	30	50	
	A0-A15, R/$\overline{\text{W}}$, BA, BS, LIC, AVMA, BUSY	C_{out}	–	10	15	pF
Frequency of Operation	MC6809E	f	0.1	–	1.0	MHz
(E and Q Inputs)	MC68A09E		0.1	–	1.5	
	MC68B09E		0.1	–	2.0	
Hi-Z (Off State) Input Current	D0-D7	I_{TSI}	–	2.0	10	μA
(V_{in} = 0.4 to 2.4 V, V_{CC} = max)	A0-A15, R/$\overline{\text{W}}$		–	–	100	

*Capacitances are periodically tested rather than 100% tested.

BUS TIMING CHARACTERISTICS (See Notes 1, 2, 3, and 4)

Ident. Number	Characteristics	Symbol	MC6809E		MC68A09E		MC68B09E		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	450	9500	295	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	450	9500	280	9500	220	9500	ns
7	Delay Time, E to Q Rise	t_{EQ1}	200	–	130	–	100	–	ns
7A	Delay Time, Q High to E Rise	t_{EQ2}	200	–	130	–	100	–	ns
7B	Delay Time, E High to Q Fall	t_{EQ3}	200	–	130	–	100	–	ns
7C	Delay Time, Q High to E Fall	t_{EQ4}	200	–	130	–	100	–	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	ns
11	Address Delay Time from E Low (BA, BS, R/$\overline{W}$)	t_{AD}	–	200	–	140	–	110	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time	t_{ACC}	695	–	440	–	330	–	ns
30	Control Delay Time	t_{CD}	–	300	–	250	–	200	ns
	Interrupts, $\overline{HALT}$, $\overline{RESET}$, and TSC Setup Time (Figures 6, 7, 8, 9, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	TSC Drive to Valid Logic Level (Figure 13)	t_{TSV}	–	210	–	150	–	120	ns
	TSC Release MOS Buffers to High Impedance (Figure 13)	t_{TSR}	–	200	–	140	–	110	ns
	TSC Hi-Z Delay Time (Figure 13)	t_{TSD}	–	120	–	85	–	80	ns
	Processor Control Rise and Fall Time (Figure 7)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

FIGURE 1 – READ/WRITE DATA TO MEMORY OR PERIPHERALS TIMING DIAGRAM



NOTES.

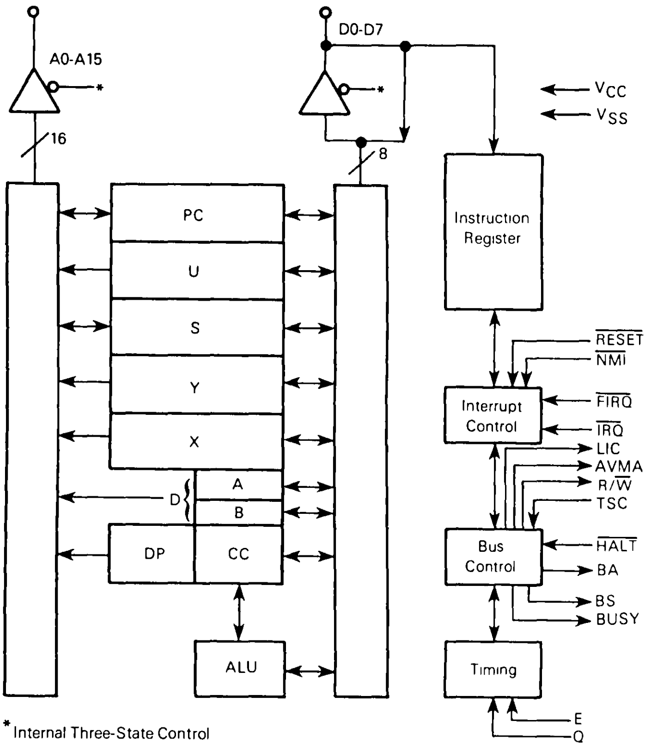
1 Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified

2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

3 Hold time (⑨) for BA and BS is not specified

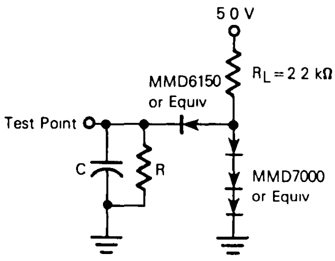
4 Usable access time is computed by 1 – 4 – 11 max – 17

FIGURE 2 — EXPANDED BLOCK DIAGRAM


A0-A15
16
D0-D7
8
VCC
VSS
PC
U
S
Y
X
D
A
B
DP
CC
ALU
Instruction Register
RESET
NMI
Interrupt Control
FIRQ
IRQ
LIC
AVMA
R/W
TSC
Bus Control
HALT
BA
BS
BUSY
Timing
E
Q

* Internal Three-State Control

FIGURE 3 — BUS TIMING TEST LOAD


5.0 V
MMD6150 or Equiv
$R_L = 2.2$ kΩ
Test Point
C
R
MMD7000 or Equiv

C = 30 pF for BA, BS, LIC, AVMA, BUSY
130 pF for D0-D7
90 pF for A0-A15, $R/\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, $R/\overline{W}$
24 kΩ for BA, BS, LIC, AVMA, BUSY

PROGRAMMING MODEL

As shown in Figure 4, the MC6809E adds three registers to the set available in the MC6800. The added registers include a direct page register, the user stack pointer, and a second index register.

ACCUMULATORS (A, B, D)

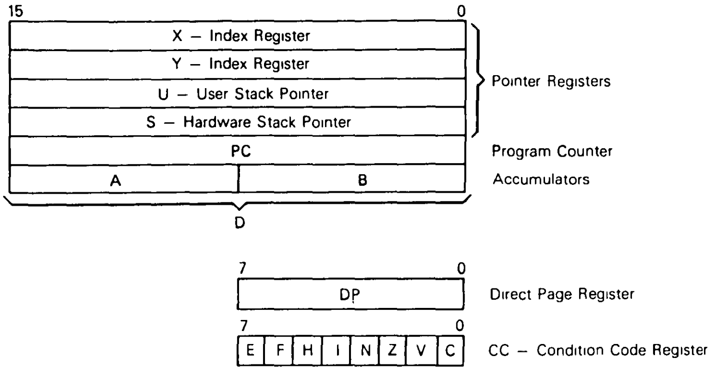
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D register, and is formed with the A register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The direct page register of the MC6809E serves to enhance the direct addressing mode. The content of this register appears at the higher address outputs (A8-A15) during direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure M6800 compatibility, all bits of this register are cleared during processor reset.

FIGURE 4 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT


15
0
X — Index Register
Y — Index Register
U — User Stack Pointer
S — Hardware Stack Pointer
Pointer Registers
PC
Program Counter
A
B
Accumulators
D
7
0
DP
Direct Page Register
7
0
E F H I N Z V C
CC — Condition Code Register

INDEX REGISTERS (X, Y)

The index registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculation of effective addresses. This address may be used to point to data directly or may be modified by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented and decremented to point to the next item of tabular type data. All four pointer registers (X, Y, U, S) may be used as index registers.

STACK POINTER (U, S)

The hardware stack pointer (S) is used automatically by the processor during subroutine calls and interrupts. The user stack pointer (U) is controlled exclusively by the programmer. This allows arguments to be passed to and from subroutines with ease. The U register is frequently used as a stack marker. Both stack pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support push and pull instructions. This allows the MC6809E to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

NOTE

The stack pointers of the MC6809E point to the top of the stack in contrast to the MC6800 stack pointer, which pointed to the next free location on stack.

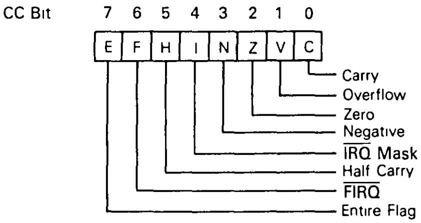
PROGRAM COUNTER

The program counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative addressing is provided allowing the program counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The condition code register defines the state of the processor at any given time. See Figure 4.

FIGURE 5 — CONDITION CODE REGISTER FORMAT


CC Bit
7 6 5 4 3 2 1 0
E F H I N Z V C
Carry
Overflow
Zero
Negative
IRQ Mask
Half Carry
FIRQ
Entire Flag

CONDITION CODE REGISTER DESCRIPTION

CARRY FLAG (C)

Bit 0 is the carry flag and is usually the carry from the binary ALU. C is also used to represent a "borrow" from subtract like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

OVERFLOW FLAG (V)

Bit 1 is the overflow flag and is set to a one by an operation which causes a signed twos complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

ZERO FLAG (Z)

Bit 2 is the zero flag and is set to a one if the result of the previous operation was identically zero.

NEGATIVE FLAG (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation Thus, a negative twos complement result will leave N set to a one

IRQ MASK (I)

Bit 4 is the IRQ mask bit. The processor will not recognize interrupts from the IRQ line if this bit is set to a one NMI, FIRQ, IRQ, RESET, and SWI all set I to a one SWI2 and SWI3 do not affect I.

HALF CARRY (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD) This bit is used by the DAA instruction to perform a BCD decimal add adjust operation The state of this flag is undefined in all subtract-like instructions

FIRQ MASK (F)

Bit 6 is the FIRQ mask bit The processor will not recognize interrupts from the FIRQ line if this bit is a one NMI, FIRQ, SWI, and RESET all set F to a one IRQ, SWI2, and SWI3 do not affect F.

ENTIRE FLAG (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the condition code register represents past action

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the address bus. When the processor does not require the bus for a data transfer, it will output address FFFF$_{16}$, R/W = 1, and BS = 0, this is a "dummy access" or VMA cycle. All address bus drivers are made high-impedance when output bus available (BA) is high or when TSC is asserted Each pin will drive one Schottky TTL load or four LSTTL loads and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus Each pin will drive one Schottky TTL load or four LSTTL loads and 130 pF

READ/WRITE (R/W)

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. R/W is made high impedance when BA is high or when TSC is asserted.

RESET

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 6. The reset vectors are fetched from locations FFFE$_{16}$ and FFFF$_{16}$ (Table 1) when interrupt acknowledge is true, (BA•BS = 1). During initial power on, the reset line should be held low until the clock input signals are fully operational.

Because the RESET pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage ensures that all peripherals are out of the reset state before the processor.

HALT

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the halt state While halted, the MPU will not respond to external real-time requests (FIRQ, IRQ) although NMI or RESET will be latched for later response During the halt state, Q and E should continue to run normally A halted state (BA•BS = 1) can be achieved by pulling HALT low while RESET is still low See Figure 7

BUS AVAILABLE, BUS STATUS (BA, BS)

The bus available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. When BA goes low, a dead cycle will elapse before the MPU acquires the bus BA will not be asserted when TSC is active, thus allowing dead cycle consistency.

The bus status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q)

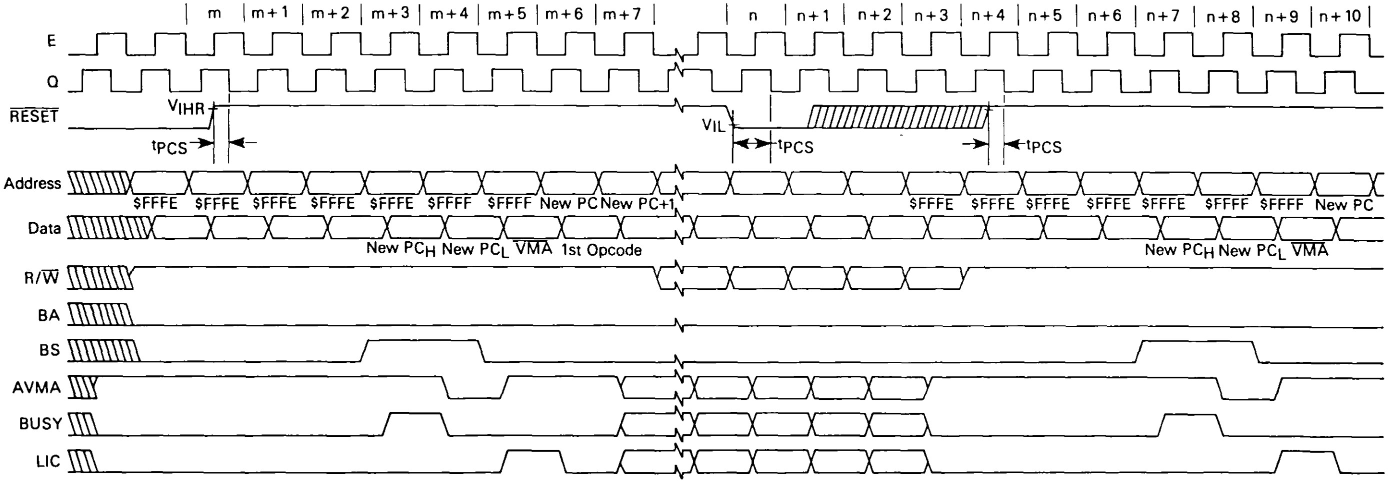
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt Acknowledge

Interrupt Acknowledge is indicated during both cycles of a hardware vector fetch (RESET, NMI, FIRQ, IRQ, SWI, SWI2, SWI3) This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device See Table 1.

TABLE 1 — MEMORY MAP FOR INTERRUPT VECTORS

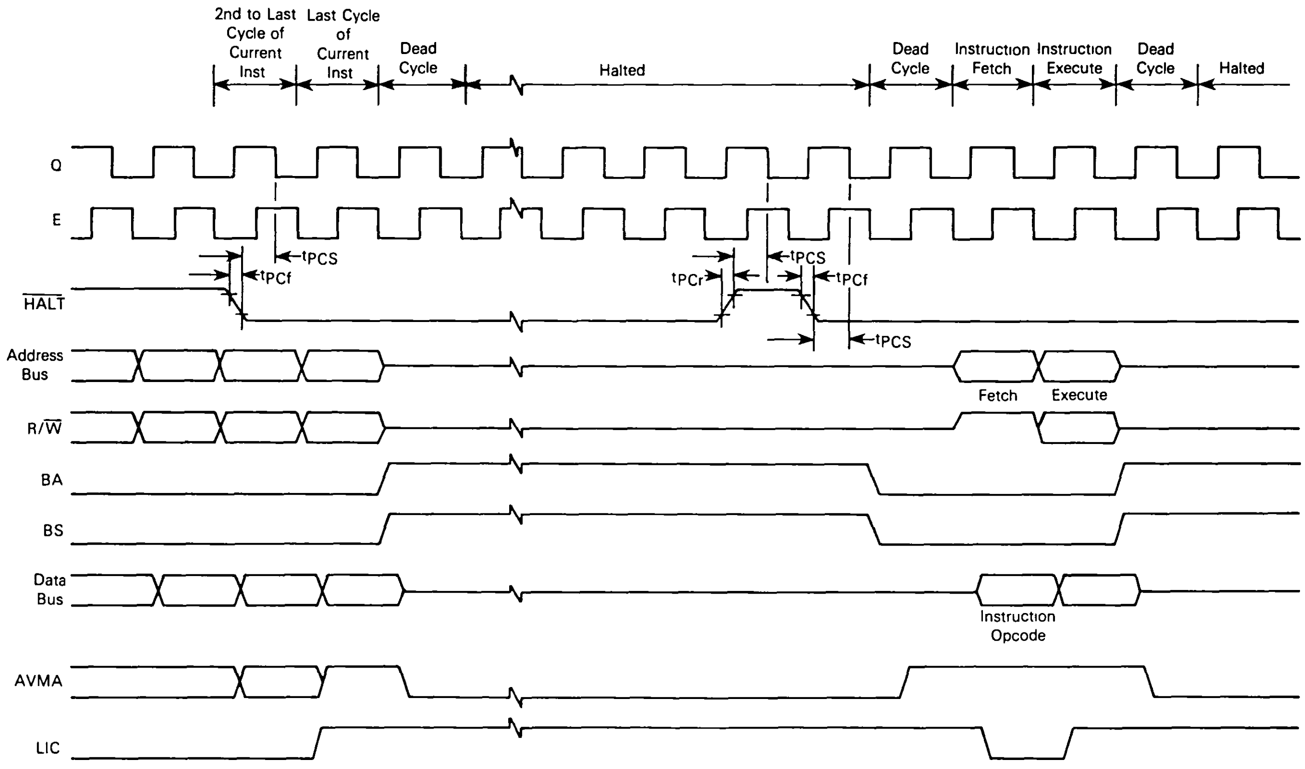
Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	RESET
FFFC	FFFD	NMI
FFFA	FFFB	SWI
FFF8	FFF9	IRQ
FFF6	FFF7	FIRQ
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

FIGURE 6 — $\overline{\text{RESET}}$ TIMING


m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
n
n+1
n+2
n+3
n+4
n+5
n+6
n+7
n+8
n+9
n+10
E
Q
RESET
V_{IHR}
V_{IL}
t_{PCS}
t_{PCS}
t_{PCS}
Address
$FFFE
$FFFE
$FFFE
$FFFE
$FFFE
$FFFF
$FFFF
New PC
New PC+1
$FFFE
$FFFE
$FFFE
$FFFE
$FFFE
$FFFF
$FFFF
New PC
Data
New PC_H
New PC_L
$\overline{\text{VMA}}$
1st Opcode
New PC_H
New PC_L
$\overline{\text{VMA}}$
R/$\overline{\text{W}}$
BA
BS
AVMA
BUSY
LIC

NOTE: Timing measurements are referenced to and from a low voltage of 0 8 volts and a high voltage of 2 0 volts, unless otherwise noted

FIGURE 7 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION TIMING FOR SYSTEM DEBUG


2nd to Last Cycle of Current Inst
Last Cycle of Current Inst
Dead Cycle
Halted
Dead Cycle
Instruction Fetch
Instruction Execute
Dead Cycle
Halted
Q
E
tPCS
tPCf
tPCr
tPCS
tPCf
tPCS
HALT
Address Bus
Fetch
Execute
R/W
BA
BS
Data Bus
Instruction Opcode
AVMA
LIC

NOTE Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

Sync Acknowledge is indicated while the MPU is waiting for external synchronization on an interrupt line.

Halt Acknowledge is indicated when the MC6809E is in a halt condition.

NON MASKABLE INTERRUPT (NMI)*

A negative transition on this input requests that a non-maskable interrupt sequence be generated. A non-maskable interrupt cannot be inhibited by the program and also has a higher priority than FIRQ, IRQ, or software interrupts. During recognition of an NMI, the entire machine state is saved on the hardware stack. After reset, an NMI will not be recognized until the first program load of the hardware stack pointer (S). The pulse width of NMI low must be at least one E cycle. If the NMI input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 8.

FAST-INTERRUPT REQUEST (FIRQ)*

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard interrupt request (IRQ) and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 9.

INTERRUPT REQUEST (IRQ)*

A low level input on this pin will initiate an interrupt request sequence provided the mask bit (I) in the CC is clear. Since IRQ stacks the entire machine state, it provides a slower response to interrupts than FIRQ. IRQ also has a lower priority than FIRQ. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 8.

CLOCK INPUTS E, Q

E and Q are the clock signals required by the MC6809E. Q must lead E, that is, a transition on Q must be followed by a similar transition on E after a minimum delay. Addresses will be valid from the MPU, t_{AD} after the falling edge of E, and data will be latched from the bus by the falling edge of E. While the Q input is fully TTL compatible, the E input directly drives internal MOS circuitry and, thus, requires a high level above normal TTL levels. This approach minimizes clock skew inherent with an internal buffer. Refer to **BUS TIMING CHARACTERISTICS** for E and Q and to Figure 10 which shows a simple clock generator for the MC6809E.

BUSY

BUSY will be high for the read and modify cycles of a read-modify-write instruction and during the access of the first byte of a double-byte operation (e.g., LDX, STD, ADDD). BUSY is also high during the first byte of any indirect or other vector fetch (e.g., jump extended, SWI indirect, etc.).

In a multiprocessor system, BUSY indicates the need to defer the rearbitration of the next bus cycle to insure the integrity of the above operations. This difference provides the indivisible memory access required for a "test-and-set" primitive, using any one of several read-modify-write instructions.

BUSY does not become active during PSH or PUL operations. A typical read-modify-write instruction (ASL) is shown in Figure 11. Timing information is given in Figure 12. BUSY is valid t_{CD} after the rising edge of Q.

AVMA

AVMA is the advanced VMA signal and indicates that the MPU will use the bus in the following bus cycle. The predictive nature of the AVMA signal allows efficient shared-bus multiprocessor systems. AVMA is low when the MPU is in either a HALT or SYNC state. AVMA is valid t_{CD} after the rising edge of Q.

LIC

LIC (last instruction cycle) is high during the last cycle of every instruction, and its transition from high to low will indicate that the first byte of an opcode will be latched at the end of the present bus cycle. LIC will be high when the MPU is halted at the end of an instruction (i.e., not in CWAI or RESET), in sync state, or while stacking during interrupts. LIC is valid t_{CD} after the rising edge of Q.

TSC

TSC (three-state control) will cause MOS address, data, and R/W buffers to assume a high-impedance state. The control signals (BA, BS, BUSY, AVMA, and LIC) will not go to the high-impedance state. TSC is intended to allow a single bus to be shared with other bus masters (processors or DMA controllers).

While E is low, TSC controls the address buffers and R/W directly. The data bus buffers during a write operation are in a high-impedance state until Q rises at which time, if TSC is true, they will remain in a high-impedance state. If TSC is held beyond the rising edge of E, then it will be internally latched, keeping the bus drivers in a high-impedance state for the remainder of the bus cycle. See Figure 13.

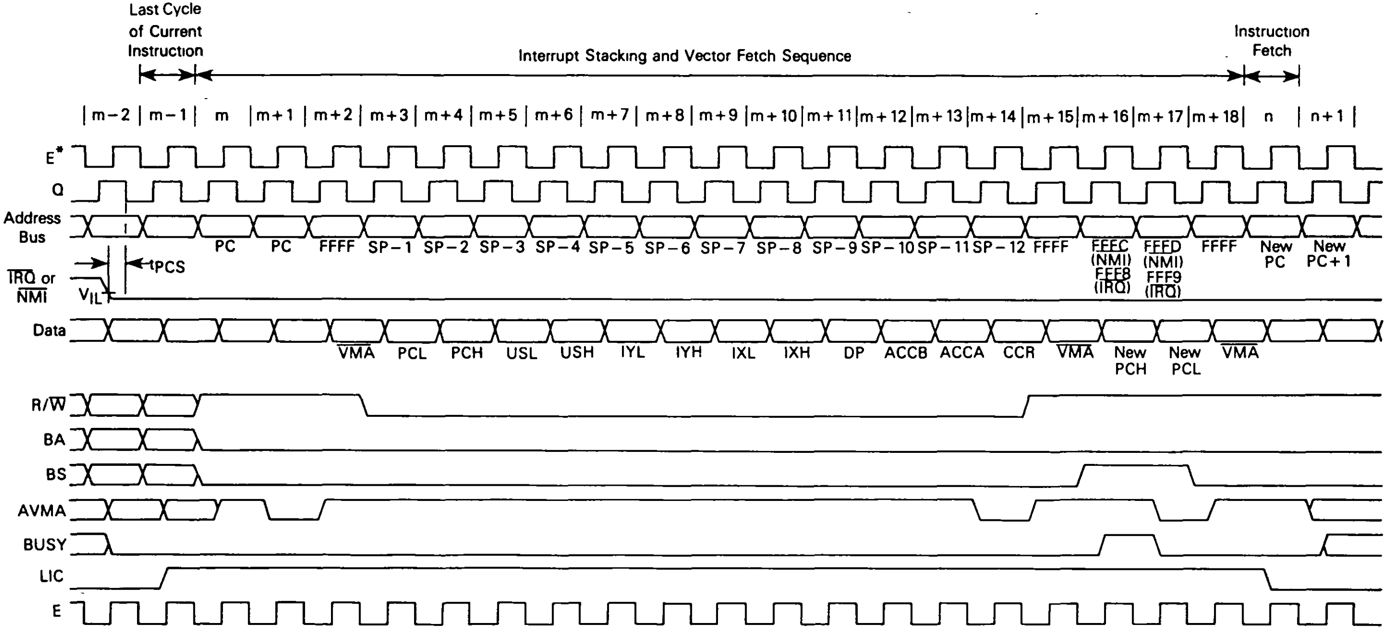
MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after RESET and is repeated indefinitely unless altered by a special instruction or hardware occurrence. Software instructions that alter normal MPU operation are: SWI, SWI2, SWI3, CWAI, RTI, and SYNC. An interrupt or HALT input can also alter the normal execution of instructions. Figure 14 is the flowchart for the MC6809E.

---

*NMI, FIRQ, and IRQ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or CWAI condition is present. If IRQ and FIRQ do not remain low until completion of the current instruction, they may not be recognized. However, NMI is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of RESET and the rising edge of BS indicating RESET acknowledge. See RESET sequence in the MPU flowchart in Figure 14.

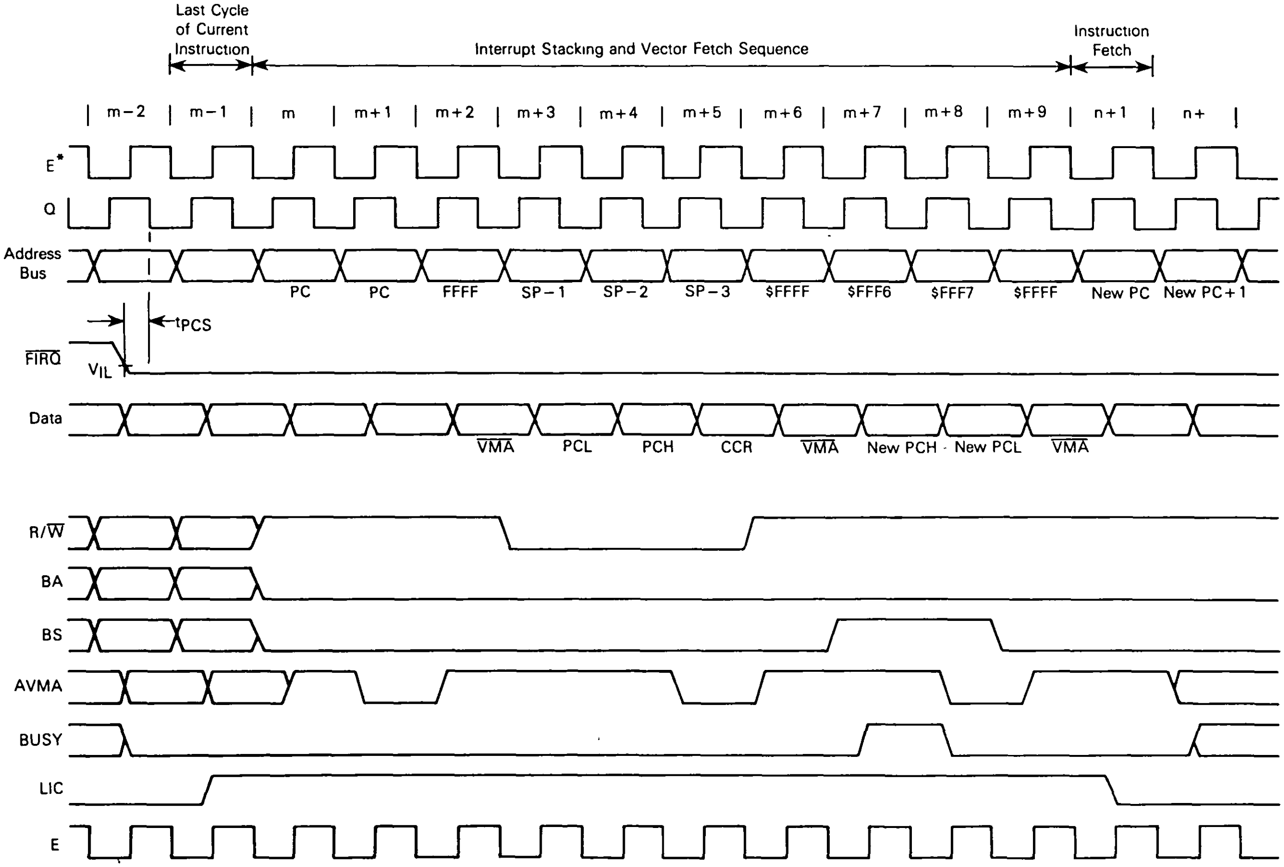
FIGURE 8 — $\overline{\text{IRQ}}$ AND $\overline{\text{NMI}}$ INTERRUPT TIMING


Last Cycle of Current Instruction
Interrupt Stacking and Vector Fetch Sequence
Instruction Fetch
m−2
m−1
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
m+10
m+11
m+12
m+13
m+14
m+15
m+16
m+17
m+18
n
n+1
E*
Q
Address Bus
PC
PC
FFFF
SP−1
SP−2
SP−3
SP−4
SP−5
SP−6
SP−7
SP−8
SP−9
SP−10
SP−11
SP−12
FFFF
FFFC (NMI) FFF8 (IRQ)
FFFD (NMI) FFF9 (IRQ)
FFFF
New PC
New PC+1
IRQ or NMI
VIL
tPCS
Data
VMA
PCL
PCH
USL
USH
IYL
IYH
IXL
IXH
DP
ACCB
ACCA
CCR
VMA
New PCH
New PCL
VMA
R/W
BA
BS
AVMA
BUSY
LIC
E

* E clock shown for reference only

NOTE: Timing measurements are referenced to and from a low voltage of 0 8 volts and a high voltage of 2.0 volts, unless otherwise noted

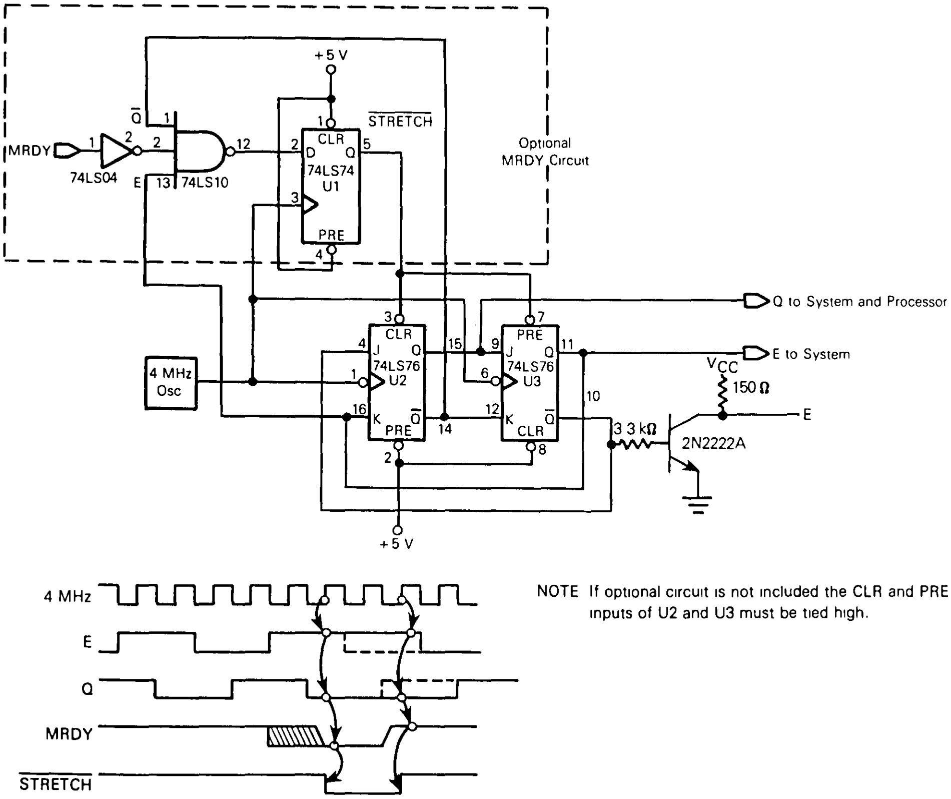
FIGURE 9 — $\overline{\text{FIRQ}}$ INTERRUPT TIMING


Last Cycle of Current Instruction
Interrupt Stacking and Vector Fetch Sequence
Instruction Fetch
m−2
m−1
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
n+1
n+
E*
Q
Address Bus
PC
PC
FFFF
SP−1
SP−2
SP−3
$FFFF
$FFF6
$FFF7
$FFFF
New PC
New PC+1
tPCS
FIRQ
VIL
Data
VMA
PCL
PCH
CCR
VMA
New PCH
New PCL
VMA
R/W
BA
BS
AVMA
BUSY
LIC
E

* E clock shown for reference only

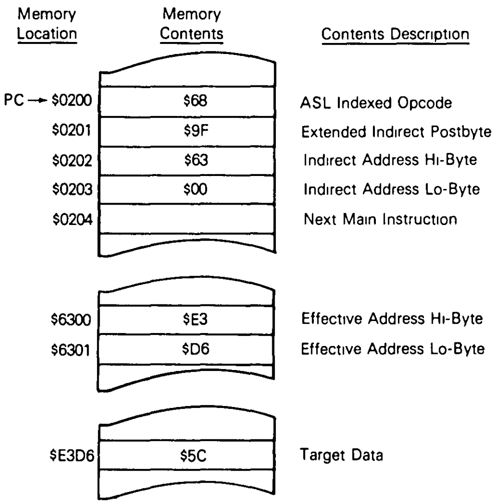
NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2 0 volts, unless otherwise noted

FIGURE 10 — CLOCK GENERATOR


+5 V
Q̄
MRDY
74LS04
74LS10
E
STRETCH
CLR
D
Q
74LS74
U1
PRE
Optional
MRDY Circuit
Q to System and Processor
E to System
4 MHz
Osc
74LS76
U2
J
K
Q̄
74LS76
U3
VCC
150 Ω
E
3.3 kΩ
2N2222A
4 MHz
E
Q
MRDY
STRETCH

NOTE If optional circuit is not included the CLR and PRE inputs of U2 and U3 must be tied high.

FIGURE 11 — READ-MODIFY-WRITE INSTRUCTION EXAMPLE (ASL EXTENDED INDIRECT)



Memory Location	Memory Contents	Contents Description
PC → $0200	$68	ASL Indexed Opcode
$0201	$9F	Extended Indirect Postbyte
$0202	$63	Indirect Address Hi-Byte
$0203	$00	Indirect Address Lo-Byte
$0204		Next Main Instruction
$6300	$E3	Effective Address Hi-Byte
$6301	$D6	Effective Address Lo-Byte
$E3D6	$5C	Target Data

FIGURE 12 — BUSY TIMING

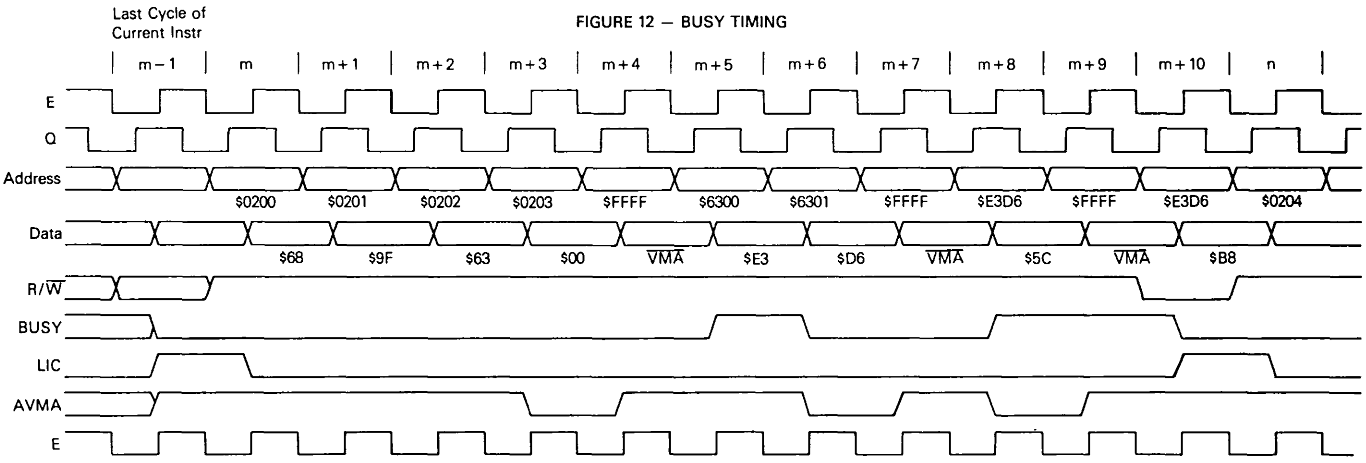
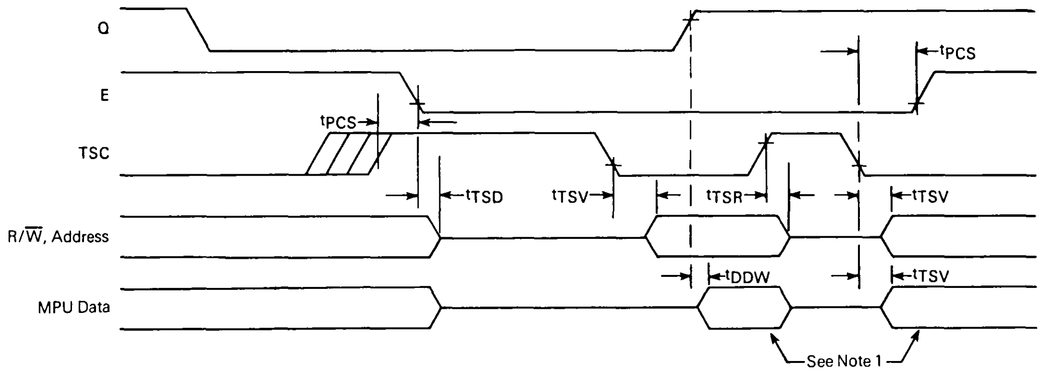

Last Cycle of Current Instr
m−1
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
m+9
m+10
n
E
Q
Address
$0200
$0201
$0202
$0203
$FFFF
$6300
$6301
$FFFF
$E3D6
$FFFF
$E3D6
$0204
Data
$68
$9F
$63
$00
$\overline{VMA}$
$E3
$D6
$\overline{VMA}$
$5C
$\overline{VMA}$
$B8
R/$\overline{W}$
BUSY
LIC
AVMA
E

FIGURE 13 — TSC TIMING


Q
E
TSC
R/$\overline{W}$, Address
MPU Data
tPCS
tPCS
tTSD
tTSV
tTSR
tTSV
tDDW
tTSV
See Note 1

NOTES

1. Data will be asserted by the MPU only during the interval while R/$\overline{W}$ is low and (E or Q) is high. A composite bus cycle is shown to give most cases of timing.
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809E has the most complete set of addressing modes available on any microcomputer. For example, the MC6809E has 59 basic instructions, however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the MC6809E:

- Inherent (Includes Accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary Examples of inherent addressing are ABX, DAA, SWI, ASRA, and CLRB

IMMEDIATE ADDRESSING

In immediate addressing, the effective address of the data is the location immediately following the opcode (i e , the data to be used in the instruction immediately following the opcode of the instruction) The MC6809E uses both 8- and 16-bit immediate values depending on the size of argument specified by the opcode Examples of instructions with immediate addressing are

```
LDA   #$20
LDX   #$F000
LDY   #CAT
```

NOTE

signifies immediate addressing, $ signifies hexadecimal value to the MC6809 assembler.

EXTENDED ADDRESSING

In extended addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction Note that the address generated by an extended instruction defines an absolute address and is not position independent. Examples of extended addressing include.

```
LDA   CAT
STX   MOUSE
LDD   $2000
```

EXTENDED INDIRECT

As a special case of indexed addressing (discussed below), one level of indirection may be added to extended addressing In extended indirect, the two bytes following the postbyte of an indexed instruction contain the address of the data

```
LDA   [CAT]
LDX   [$FFFE]
STU   [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode This byte specifies the lower eight bits of the address to be used. The upper eight bits of the address are supplied by the direct page register Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register Since the DP register is set to $00 on reset, direct addressing on the MC6809E is upward compatible with direct addressing on the M6800 Indirection is not allowed in direct addressing. Some examples of direct addressing are.

```
LDA   where DP=$00
LDB   where DP=$10
LDD   <CAT
```

NOTE

< is an assembler directive which forces direct addressing.

REGISTER ADDRESSING

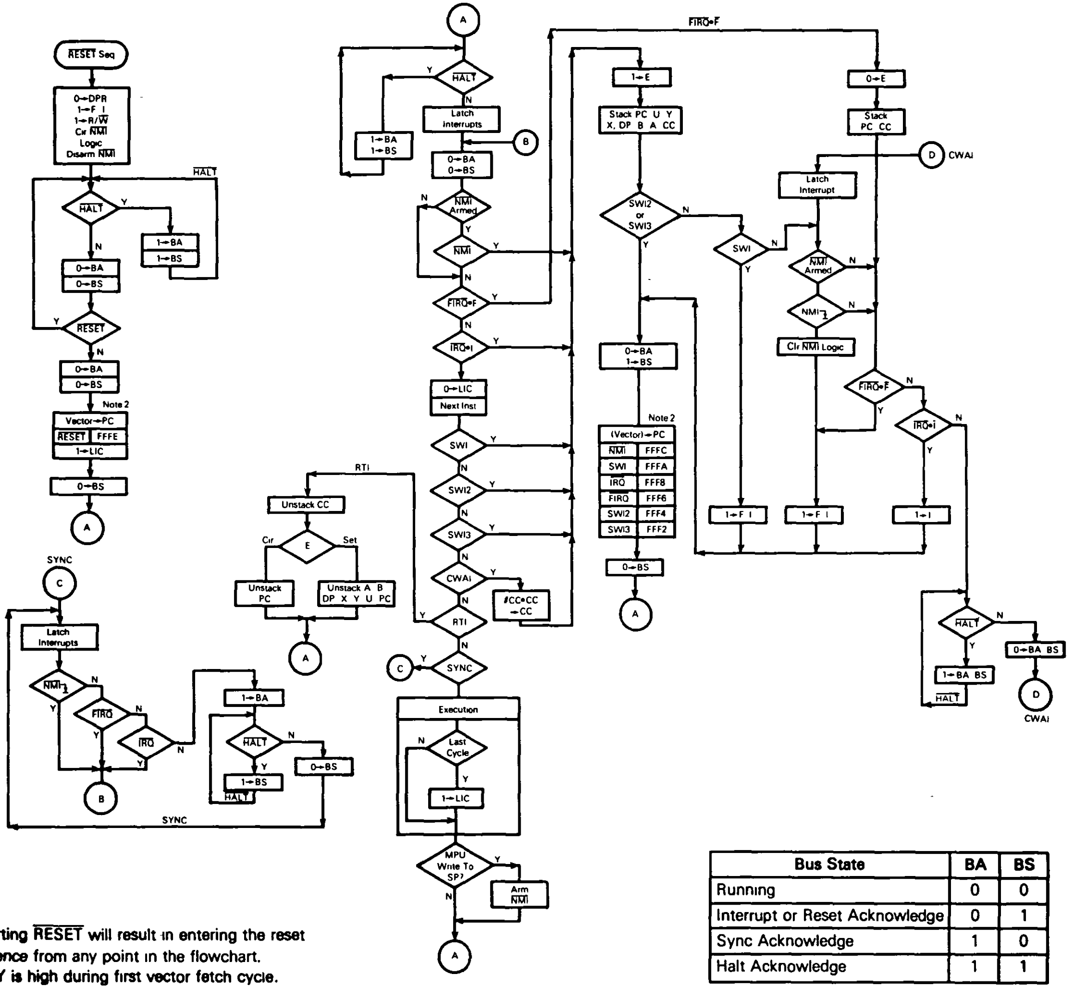
Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction This is called a postbyte Some examples of register addressing are

TFR	X, Y	Transfers X into Y
EXG	A, B	Exchanges A with B
PSHS	A, B, X, Y	Push Y, X, B and A onto S stack
PULU	X, Y, D	Pull D, X, and Y from U stack

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode, as well as the pointer register to be used Figure 15 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation

FIGURE 14 — FLOWCHART FOR MC6809E INSTRUCTIONS


RESET Seq
0→DPR
1→F I
1→R/W
Clr NMI Logic
Disarm NMI
HALT
Y
N
1→BA
1→BS
0→BA
0→BS
RESET
Note 2
Vector→PC
RESET FFFE
1→LIC
0→BS
A
SYNC
C
Latch Interrupts
NMI
FIRQ
IRQ
B
HALT
0→BS
1→BS
A
Latch Interrupts
B
NMI Armed
NMI
FIRQ•F
IRQ•I
0→LIC
Next Inst
SWI
SWI2
SWI3
CWAI
#CC•CC→CC
RTI
Unstack CC
E
Clr
Set
Unstack PC
Unstack A B DP X Y U PC
SYNC
Execution
Last Cycle
1→LIC
MPU Write To SP?
Arm NMI
1→E
Stack PC U Y X, DP B A CC
SWI2 or SWI3
0→BA
1→BS
(Vector)→PC
NMI FFFC
SWI FFFA
IRQ FFF8
FIRQ FFF6
SWI2 FFF4
SWI3 FFF2
0→E
Stack PC CC
D
CWAI
Latch Interrupt
Clr NMI Logic
1→F I
1→I
0→BA BS
1→BA BS

Bus State	BA	BS
Running	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt Acknowledge	1	1

NOTES: 1. Asserting $\overline{\text{RESET}}$ will result in entering the reset sequence from any point in the flowchart.
2. BUSY is high during first vector fetch cycle.

FIGURE 15 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit 7	6	5	4	3	2	1	0	Indexed Addressing Mode
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R+ +
1	R	R	0	0	0	1	0	,− R
1	R	R	i	0	0	1	1	,− − R
1	R	R	i	0	1	0	0	EA = ,R + 0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R + D Offset
1	x	x	i	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]

Addressing Mode Field

Indirect Field
(Sign Bit when b7 = 0)

Register Field RR
00 = X
01 = Y
10 = U
11 = S

x = Don't Care
d = Offset Bit
i = 0 = Not Indirect
1 = Indirect

ZERO-OFFSET INDEXED — In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

LDD 0, X
LDA ,S

CONSTANT OFFSET INDEXED — In this mode, a two's-complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offset are available

5-bit (−16 to +15)
8-bit (−128 to +128)
16-bit (−32768 to +32767)

The twos complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles The twos complement 8-bit offset is contained in a single byte following the postbyte. The twos complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be cocerned with the size of this offset since the assembler will select the optimal size automatically

Examples of constant-offset indexing are:

LDA 23,X
LDX −2,S
LDY 300,X
LDU CAT,Y

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect				Indirect			
		Assembler Form	Postbyte Opcode	+ ~	+ #	Assembler Form	Postbyte Opcode	+ ~	+ #
Constant Offset From R (2s Complement Offsets)	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R (2s Complement Offsets)	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R+ +	1RR00001	3	0	[,R+ +]	1RR10001	6	0
	Decrement By 1	,− R	1RR00010	2	0	not allowed			
	Decrement By 2	,− − R	1RR00011	3	0	[,− − R]	1RR10011	6	0
Constant Offset From PC (2s Complement Offsets)	8-Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
	16-Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16-Bit Address	—	—	—	—	[n]	10011111	5	2

R = X, Y, U or S
x = Don't Care

RR·
00 = X
01 = Y
10 = U
11 = S

+ ~ and + # indicate the number of additional cycles and bytes respectively for the particular indexing variation

ACCUMULATOR-OFFSET INDEXED — This mode is similar to constant offset indexed except that the twos complement value in one of the accumulators (A, B, or D) and the contents of one of the pointer registers are added to form the effective address of the operand The contents of both the accumulator and the pointer register are unchanged by the addition The postbyte specifies which accumulator to use as an offset and no additional bytes are required The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are.

```
LDA   B, Y
LDX   D, Y
LEAX  B, X
```

AUTO INCREMENT/DECREMENT INDEXED — In the auto increment addressing mode, the pointer register contains the address of the operand Then, after the pointer register is used, it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or creating software stacks In auto decrement, the pointer register is decremented prior to use as the address of the data The use of auto decrement is similar to that of auto increment, but the tables, etc , are scanned from the high to low addresses The size of the increment/decrement can be either one or two to allow for tables of either 8- or 16-bit data to be accessed and is selectable by the programmer The pre-decrement, post-increment nature of these modes allows them to be used to create additional software stacks that behave identically to the U and S stacks

Some examples of the auto increment/decrement addressing modes are·

```
LDA  ,X+
STD  ,Y++
LDB  ,-Y
LDX  ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address

Consider the following instruction

STX 0,X++ (X initialized to 0)

The desired result is to store a zero in locations $0000 and $0001, then increment X to point to $0002 In reality, the following occurs

0→temp	calculate the EA, temp is a holding register
X+2→X	perform auto increment
X→(temp)	do store operation

INDEXED INDIRECT

All of the indexing modes, with the exception of auto increment/decrement by one or a ±5-bit offset, may have an additional level of indirection specified In indirect addressing, the effective address is contained at the location specified by the contents of the index register plus any offset In the example below, the A accumulator is loaded indirectly using an effective address calculated from the index register and an offset

Before Execution
A = XX (don't care)
X = $F000

```
$0100   LDA [$10,X]    EA is now $F010

$F010   $F1            $F150 is now the
$F011   $50            new EA

$F150   $AA
```

After Execution
A = $AA (actual data loaded)
X = $F000

All modes of indexed indirect are included except those which are meaningless (e g , auto increment/decrement by 1 indirect) Some examples of indexed indirect are

```
LDA  [,X]
LDD  [10,S]
LDA  [B,Y]
LDD  [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter. If the branch condition is true, then the calculated address (PC + signed offset) is loaded into the program counter Program execution continues at the new location as indicated by the PC, short (one byte offset) and long (two bytes offset) relative addressing modes are available All of memory can be reached in long relative addressing as an effective address interpreted modulo 2^{16} Some examples of relative addressing are.

```
        BEQ    CAT      (short)
        BGT    DOG      (short)
CAT     LBEQ   RAT      (long)
DOG     LBGT   RABBIT   (long)
        •
        •
        •
RAT     NOP
RABBIT  NOP
```

PROGRAM COUNTER RELATIVE

The PC can be used as the pointer register with 8- or 16-bit signed offsets. As in relative addressing, the offset is added to the current PC to create the effective address The effective address is then used as the address of the operand or data. Program counter relative addressing is used for writing position independent programs Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the program counter Examples are

```
LDA   CAT, PCR
LEAX  TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available.

```
LDA  [CAT, PCR]
LDU  [DOG, PCR]
```

INSTRUCTION SET

The instruction set of the MC6809E is similar to that of the MC6800 and is upward compatible at the source code level The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464

Some of the new instructions are described in detail below.

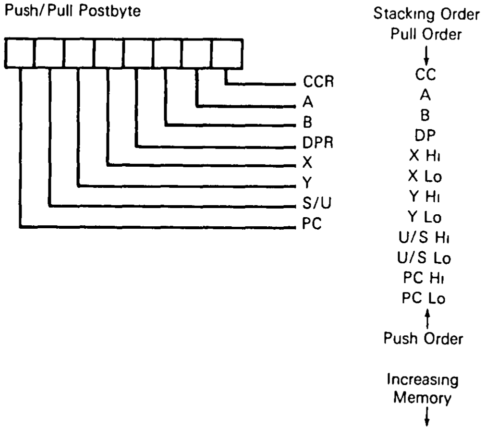
PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register or set of registers with a single instruction.

PULU/PULS

The pull instructions have the same capability of the push instruction, in reverse order. The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled. The actual push/pull sequence is fixed, each bit defines a unique register to push or pull, as shown below.

Push/Pull Postbyte



TFR/EXG

Within the MC6809E, any register may be transferred to or exchanged with another of like size; i.e., 8-bit to 8-bit or 16-bit to 16-bit. Bits 4-7 of postbyte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

Transfer/Exchange Postbyte

Source
Destination

Register Field

0000 = D (A B)
0001 = X
0010 = Y
0011 = U
0100 = S
0101 = PC
1000 = A
1001 = B
1010 = CCR
1011 = DPR

NOTE

All other combinations are undefined and INVALID

LEAX/LEAY/LEAU/LEAS

The LEA (load effective address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register This makes all the features of the internal addressing hardware available to the programmer Some of the implications of this instruction are illustrated in Table 3

The LEA instruction also allows the user to access data and tables in a position independent manner For example

```
        LEAX   MSG1, PCR
        LBSR   PDATA (Print message routine)
        •
        •
MSG1    FCC    'MESSAGE'
```

This sample program prints 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1. This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the auto increment and auto decrement addressing modes due to the sequence of internal operations The LEA internal sequence is outlined as follows.

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U, or S may be substituted for a and b.)

1. b → temp (calculate the EA)
2. b + 1 → b (modify b, postincrement)
3. temp → a (load a)

LEAa , − b

1. b − 1 → temp (calculate EA with predecrement)
2 b − 1 → b (modify b, predecrement)
3 temp → a (load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-Bit Constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-Bit Constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-Bit A Accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-Bit D Accumulator to Y
LEAU − 10, U	U − 10 → U	Substracts 10 from U
LEAS − 10, S	S − 10 → S	Used to Reserve Area on Stack
LEAS 10, S	S + 10 → S	Used to 'Clean Up' Stack
LEAX 5, S	S + 5 → X	Transfers As Well As Adds

Auto increment-by-two and auto decrement-by-two instructions work similarly Note that LEAX ,X+ does not change X, however LEAX, −X does decrement X LEAX 1,X should be used to increment X by one

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator This unsigned multiply also allows multiple-precision multiplications.

LONG AND SHORT RELATIVE BRANCHES

The MC6809E has the capability of program counter relative branching throughout the entire memory map In this mode, if the branch is to be taken, the 8- or 16-bit signed offset is added to the value of the program counter to be used as the effective address This allows the program to branch anywhere in the 64K memory map Position independent code can be easily generated through the use of relative branching Both short (8 bit) and long (16 bit) branches are available

SYNC

After encountering a sync instruction, the MPU enters a sync state, stops processing instructions, and waits for an interrupt If the pending interrupt is non-maskable ($\overline{\mathrm{NMI}}$) or maskable ($\overline{\mathrm{FIRQ}}$, $\overline{\mathrm{IRQ}}$) with its mask bit (F or I) clear, the processor will clear the sync state and perform the normal interrupt stacking and service routine Since $\overline{\mathrm{FIRQ}}$ and $\overline{\mathrm{IRQ}}$ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken If the pending interrupt is maskable ($\overline{\mathrm{FIRQ}}$, $\overline{\mathrm{IRQ}}$) with its mask bit (F or I) set, the processor will clear the sync state and continue processing by executing the next in-line instruction Figure 16 depicts sync timing

SOFTWARE INTERRUPTS

A software interrupt is an instruction which will cause an interrupt and its associated vector fetch. These software interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems Three levels of SWI are available on the MC6809E and are prioritized in the following order: SWI, SWI2, SWI3

16-BIT OPERATION

The MC6809E has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes, and pulls

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart (Figure 16) illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the MC6809E. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flowchart $\overline{\mathrm{VMA}}$ is an indication of FFFF$_{16}$ on the address bus, R/$\overline{\mathrm{W}}$ = 1 and BS = 0. The following examples illustrate the use of the chart

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                           •
                           •
                           •
$8000                      LBSR     CAT
                           •
                           •
                           •
$A000       CAT            •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{\mathrm{W}}$	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	$\overline{\mathrm{VMA}}$ Cycle
5	FFFF	*	1	$\overline{\mathrm{VMA}}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{\mathrm{VMA}}$ Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000      DEC      $A000
$A000      FCB      $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/$\overline{\mathrm{W}}$	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	$\overline{\mathrm{VMA}}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{\mathrm{VMA}}$ Cycle
7	FFFF	7F	0	Store the Decremented Data

*The data bus has the data at that particular address

INSTRUCTION SET TABLES

The instructions of the MC6809E have been broken down into five different categories They are as follows·

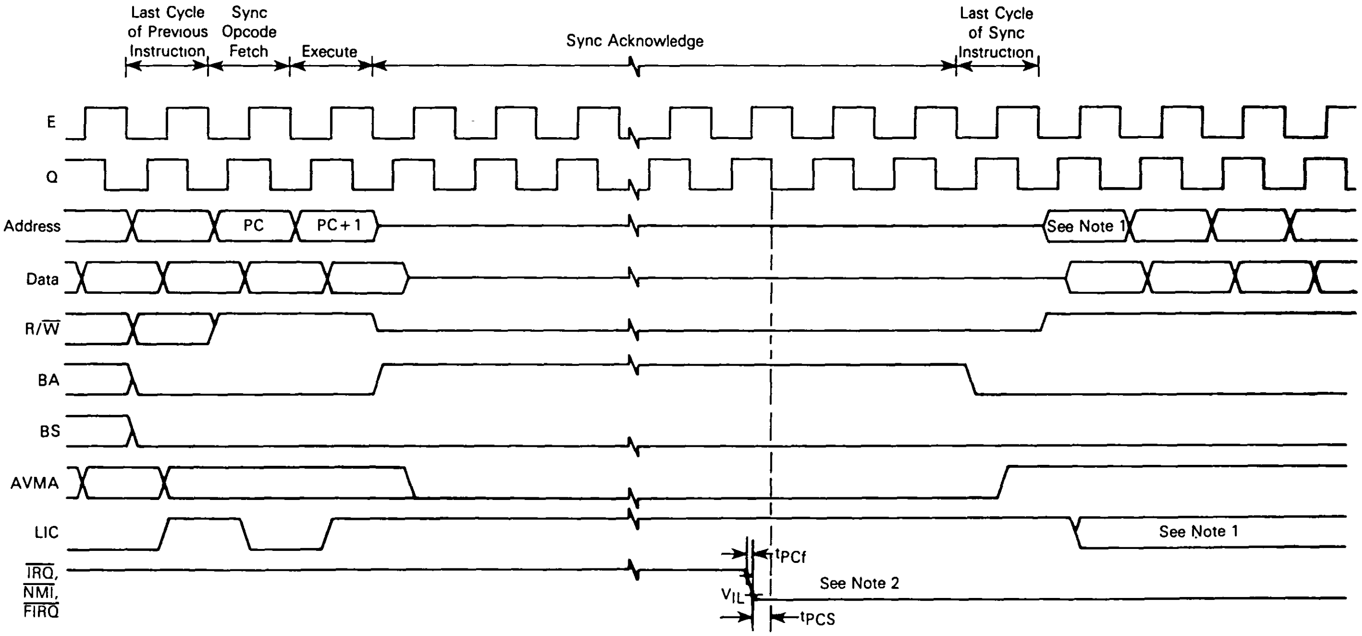
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 18 contains a compilation of data that will assist you in programming the MC6809E.

FIGURE 16 — SYNC TIMING

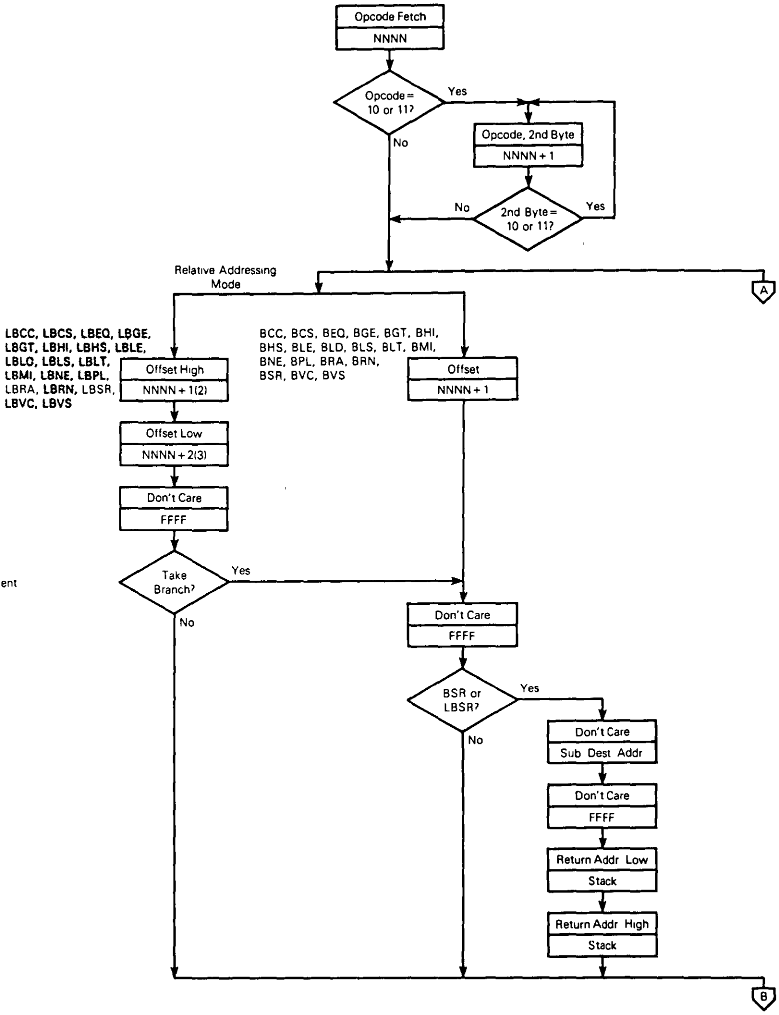

Last Cycle of Previous Instruction
Sync Opcode Fetch
Execute
Sync Acknowledge
Last Cycle of Sync Instruction
E
Q
Address
PC
PC+1
See Note 1
Data
R/W
BA
BS
AVMA
LIC
See Note 1
IRQ, NMI, FIRQ
tPCf
VIL
See Note 2
tPCS

NOTES 1. If the associated mask bit is set when the interrupt is requested, LIC will go low and this cycle will be an instruction fetch from address location PC+1. However, if the interrupt is accepted ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) LIC will remain high and interrupt processing will start with this cycle as in on Figures 8 and 9 (Interrupt Timing)

2. If mask bits are clear, $\overline{IRQ}$ and $\overline{FIRQ}$ must be held low for three cycles to guarantee that interrupt will be taken, although only one cycle is necessary to bring the processor out of SYNC

3 Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 5)


Opcode Fetch
NNNN
Opcode = 10 or 11?
Yes
No
Opcode, 2nd Byte
NNNN + 1
2nd Byte = 10 or 11?
No
Yes
Relative Addressing Mode
A
LBCC, LBCS, LBEQ, LBGE, LBGT, LBHI, LBHS, LBLE, LBLO, LBLS, LBLT, LBMI, LBNE, LBPL, LBRA, LBRN, LBSR, LBVC, LBVS
BCC, BCS, BEQ, BGE, BGT, BHI, BHS, BLE, BLO, BLS, BLT, BMI, BNE, BPL, BRA, BRN, BSR, BVC, BVS
Offset High
NNNN + 1(2)
Offset
NNNN + 1
Offset Low
NNNN + 2(3)
Don't Care
FFFF
Take Branch?
Yes
No
Don't Care
FFFF
BSR or LBSR?
Yes
No
Don't Care
Sub Dest Addr
Don't Care
FFFF
Return Addr Low
Stack
Return Addr High
Stack
B

NOTES

1. Each state shows

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2. Address NNNN is location of opcode
3. If opcode is a two byte opcode subsequent addresses are in parenthesis (—)
4. Two-byte opcodes are highlighted

FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 5)

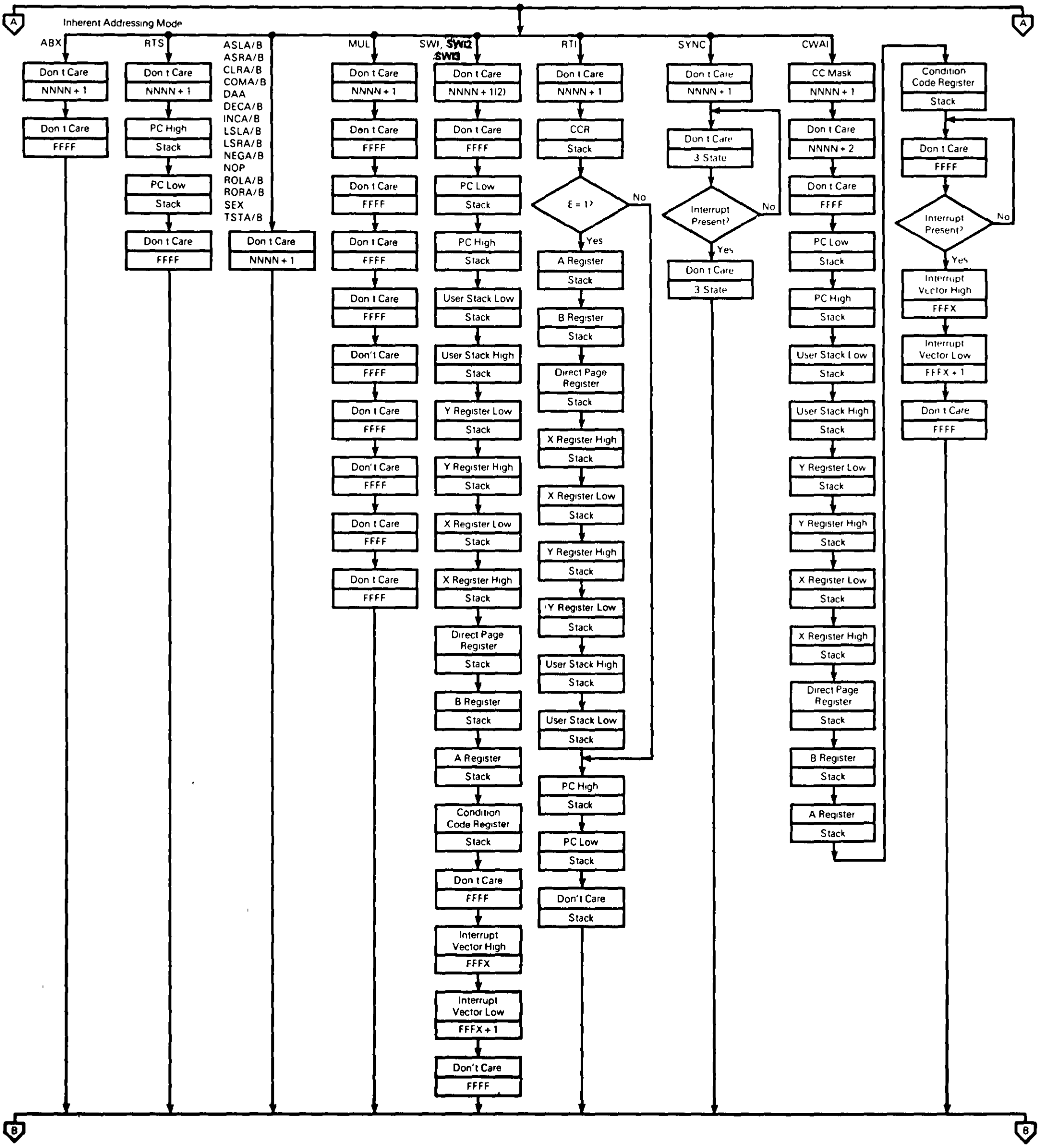

A
Inherent Addressing Mode
ABX
Don't Care
NNNN + 1
Don't Care
FFFF
RTS
Don't Care
NNNN + 1
PC High
Stack
PC Low
Stack
Don't Care
FFFF
ASLA/B
ASRA/B
CLRA/B
COMA/B
DAA
DECA/B
INCA/B
LSLA/B
LSRA/B
NEGA/B
NOP
ROLA/B
RORA/B
SEX
TSTA/B
Don't Care
NNNN + 1
MUL
Don't Care
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
SWI, SWI2
SWI3
Don't Care
NNNN + 1(2)
Don't Care
FFFF
PC Low
Stack
PC High
Stack
User Stack Low
Stack
User Stack High
Stack
Y Register Low
Stack
Y Register High
Stack
X Register Low
Stack
X Register High
Stack
Direct Page Register
Stack
B Register
Stack
A Register
Stack
Condition Code Register
Stack
Don't Care
FFFF
Interrupt Vector High
FFFX
Interrupt Vector Low
FFFX + 1
Don't Care
FFFF
RTI
Don't Care
NNNN + 1
CCR
Stack
E = 1?
No
Yes
A Register
Stack
B Register
Stack
Direct Page Register
Stack
X Register High
Stack
X Register Low
Stack
Y Register High
Stack
Y Register Low
Stack
User Stack High
Stack
User Stack Low
Stack
PC High
Stack
PC Low
Stack
Don't Care
Stack
SYNC
Don't Care
NNNN + 1
Don't Care
3 State
Interrupt Present?
No
Yes
Don't Care
3 State
CWAI
CC Mask
NNNN + 1
Don't Care
NNNN + 2
Don't Care
FFFF
PC Low
Stack
PC High
Stack
User Stack Low
Stack
User Stack High
Stack
Y Register Low
Stack
Y Register High
Stack
X Register Low
Stack
X Register High
Stack
Direct Page Register
Stack
B Register
Stack
A Register
Stack
Condition Code Register
Stack
Don't Care
FFFF
Interrupt Present?
No
Yes
Interrupt Vector High
FFFX
Interrupt Vector Low
FFFX + 1
Don't Care
FFFF
A
B
B

FIGURE 17 — CYCLE-BY-CYLE PERFORMANCE (Sheet 3 of 5)

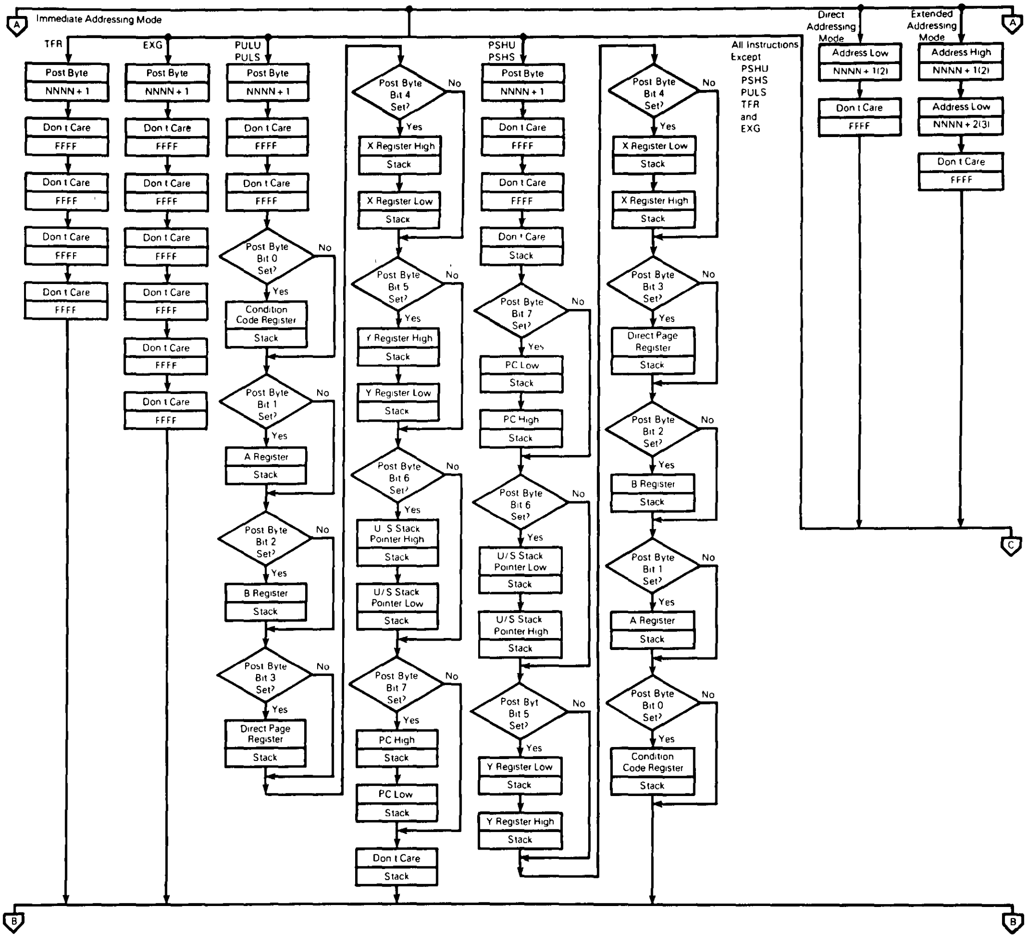
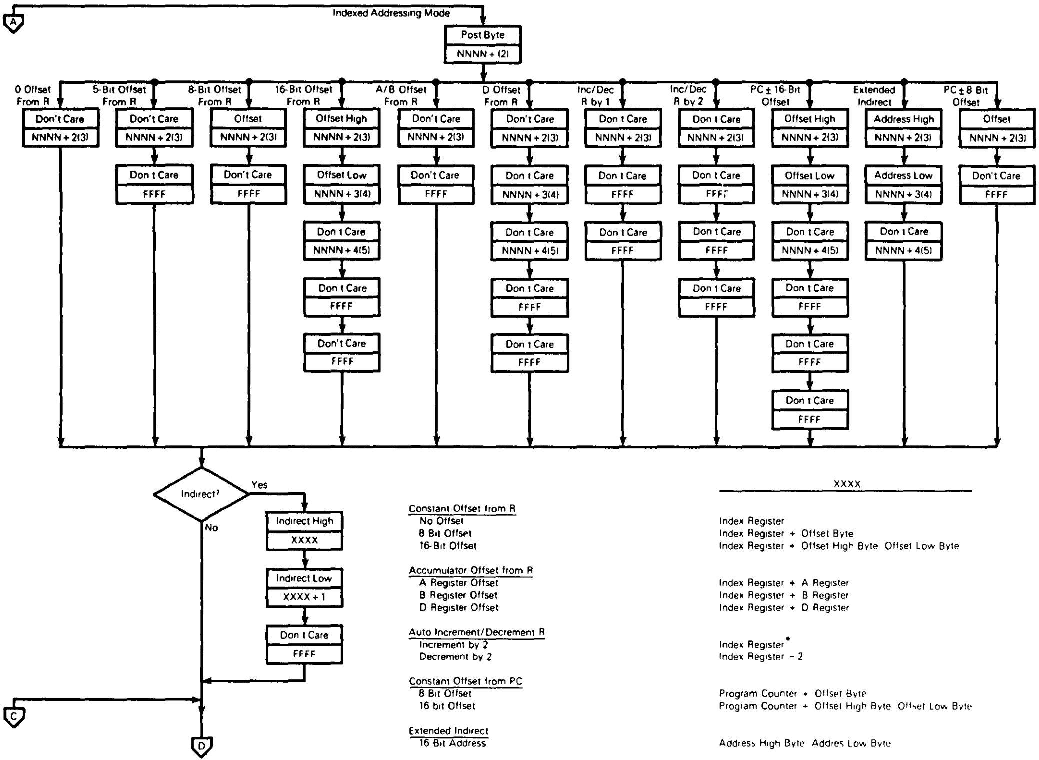

A
Immediate Addressing Mode
TFR
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
EXG
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
PULU
PULS
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Post Byte Bit 0 Set?
No
Yes
Condition Code Register
Stack
Post Byte Bit 1 Set?
No
Yes
A Register
Stack
Post Byte Bit 2 Set?
No
Yes
B Register
Stack
Post Byte Bit 3 Set?
No
Yes
Direct Page Register
Stack
Post Byte Bit 4 Set?
No
Yes
X Register High
Stack
X Register Low
Stack
Post Byte Bit 5 Set?
No
Yes
Y Register High
Stack
Y Register Low
Stack
Post Byte Bit 6 Set?
No
Yes
U/S Stack Pointer High
Stack
U/S Stack Pointer Low
Stack
Post Byte Bit 7 Set?
No
Yes
PC High
Stack
PC Low
Stack
Don't Care
Stack
PSHU
PSHS
Post Byte
NNNN + 1
Don't Care
FFFF
Don't Care
FFFF
Don't Care
Stack
Post Byte Bit 7 Set?
No
Yes
PC Low
Stack
PC High
Stack
Post Byte Bit 6 Set?
No
Yes
U/S Stack Pointer Low
Stack
U/S Stack Pointer High
Stack
Post Byte Bit 5 Set?
No
Yes
Y Register Low
Stack
Y Register High
Stack
Post Byte Bit 4 Set?
No
Yes
X Register Low
Stack
X Register High
Stack
Post Byte Bit 3 Set?
No
Yes
Direct Page Register
Stack
Post Byte Bit 2 Set?
No
Yes
B Register
Stack
Post Byte Bit 1 Set?
No
Yes
A Register
Stack
Post Byte Bit 0 Set?
No
Yes
Condition Code Register
Stack
All Instructions Except PSHU PSHS PULS TFR and EXG
Direct Addressing Mode
Address Low
NNNN + 1(2)
Don't Care
FFFF
Extended Addressing Mode
Address High
NNNN + 1(2)
Address Low
NNNN + 2(3)
Don't Care
FFFF
A
C
B
B

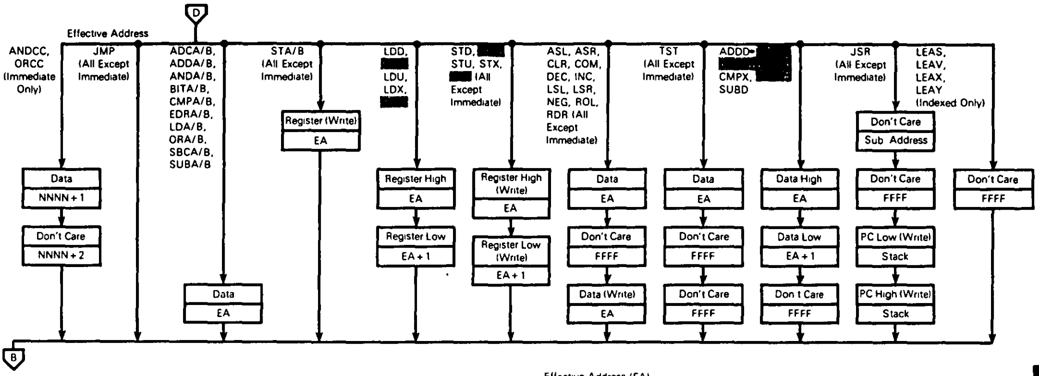
FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 5)


A
Indexed Addressing Mode
Post Byte
NNNN + (2)
0 Offset From R
Don't Care
NNNN + 2(3)
5-Bit Offset From R
Don't Care
NNNN + 2(3)
Don't Care
FFFF
8-Bit Offset From R
Offset
NNNN + 2(3)
Don't Care
FFFF
16-Bit Offset From R
Offset High
NNNN + 2(3)
Offset Low
NNNN + 3(4)
Don't Care
NNNN + 4(5)
Don't Care
FFFF
Don't Care
FFFF
A/B Offset From R
Don't Care
NNNN + 2(3)
Don't Care
FFFF
D Offset From R
Don't Care
NNNN + 2(3)
Don't Care
NNNN + 3(4)
Don't Care
NNNN + 4(5)
Don't Care
FFFF
Don't Care
FFFF
Inc/Dec R by 1
Don't Care
NNNN + 2(3)
Don't Care
FFFF
Don't Care
FFFF
Inc/Dec R by 2
Don't Care
NNNN + 2(3)
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
PC ± 16-Bit Offset
Offset High
NNNN + 2(3)
Offset Low
NNNN + 3(4)
Don't Care
NNNN + 4(5)
Don't Care
FFFF
Don't Care
FFFF
Don't Care
FFFF
Extended Indirect
Address High
NNNN + 2(3)
Address Low
NNNN + 3(4)
Don't Care
NNNN + 4(5)
PC ± 8 Bit Offset
Offset
NNNN + 2(3)
Don't Care
FFFF
Indirect?
Yes
No
Indirect High
XXXX
Indirect Low
XXXX + 1
Don't Care
FFFF
C
D

	XXXX
Constant Offset from R	
No Offset	Index Register
8 Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register − 2
Constant Offset from PC	
8 Bit Offset	Program Counter + Offset Byte
16 bit Offset	Program Counter + Offset High Byte Offset Low Byte
Extended Indirect	
16 Bit Address	Address High Byte Address Low Byte

*The index register is incremented following the indexed access

FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 5)


D
Effective Address
ANDCC, ORCC (Immediate Only)
JMP (All Except Immediate)
ADCA/B, ADDA/B, ANDA/B, BITA/B, CMPA/B, EORA/B, LDA/B, ORA/B, SBCA/B, SUBA/B
STA/B (All Except Immediate)
LDD, [redacted] LDU, LDX, [redacted]
STD, [redacted] STU, STX, [redacted] (All Except Immediate)
ASL, ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR (All Except Immediate)
TST (All Except Immediate)
ADDD, [redacted] CMPX, [redacted] SUBD
JSR (All Except Immediate)
LEAS, LEAU, LEAX, LEAY (Indexed Only)
Data
NNNN + 1
Don't Care
NNNN + 2
Data
EA
Register (Write)
EA
Register High
EA
Register Low
EA + 1
Register High (Write)
EA
Register Low (Write)
EA + 1
Data
EA
Don't Care
FFFF
Data (Write)
EA
Data
EA
Don't Care
FFFF
Don't Care
FFFF
Data High
EA
Data Low
EA + 1
Don't Care
FFFF
Don't Care
Sub Address
Don't Care
FFFF
PC Low (Write)
Stack
PC High (Write)
Stack
Don't Care
FFFF
B

	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5 Bit Offset	Index Register
8 Bit Offset	Index Register + Post Byte
16 Bit Offset	Index Register + Post Byte High Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register − 1
Decrement by 2	Index Register − 2
Constant Offset from PC	
8 Bit Offset	Program Counter + Offset Byte
16 Bit Offset	Program Counter + Offset High Byte Offset Low Byte
Direct	Direct Page Register Address Low
Extended	Address High Address Low
Immediate	NNNN + 1

*The index register is incremented following the indexed access

TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE: A, B, CC or DP may be pushed to (pulled from) either stack with PSHS, PSHU (PULS, PULU) instructions

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U or PC
TFR R, D	Transfer X, Y, S, U or PC to D

NOTE. D may be pushed (pulled) to either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, S, U or PC with D, X, Y, S, U or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 — BRANCH INSTRUCTIONS

Instruction	Description
SIMPLE BRANCHES	
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
SIGNED BRANCHES	
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2's complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2's complement result
BLT, LBLT	Branch if less than (signed)
UNSIGNED BRANCHES	
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
OTHER BRANCHES	
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 — MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	*				31	LEAY		4+	2+	61	*			
02	*				32	LEAS		4+	2+	62	*			
03	COM		6	2	33	LEAU	Indexed	4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*	—			68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX		3	1	6A	DEC		6+	2+
0B	*				3B	RTI		6/15	1	6B	*			
0C	INC		6	2	3C	CWAI		≥20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL	Inherent	11	1	6D	TST		6+	2+
0E	JMP		3	2	3E	*	—			6E	JMP		3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	—	—	—	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	—	—	—	41	*				71	*			
12	NOP	Inherent	2	1	42	*				72	*			
13	SYNC	Inherent	≥4	1	43	COMA		2	1	73	COM		7	3
14	*				44	LSRA		2	1	74	LSR		7	3
15	*				45	*				75	*			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	*				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	*	—			4B	*				7B	*			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG	Immed	8	2	4E	*				7E	JMP		4	3
1F	TFR	Immed	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN		3	2	51	*				81	CMPA		2	2
22	BHI		3	2	52	*				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	*				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	*			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	*				8B	ADDA		2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT		3	2	5E	*				8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	*			

LEGEND.

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)
# Number of program bytes
* Denotes unused opcode

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

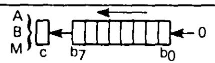
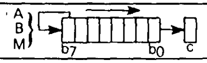
OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

NOTE All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDU	Immed	3	3
CF	*			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

OP	Mnem	Mode	~	#
Page 2 and 3 Machine Codes				
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

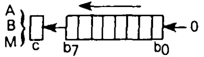
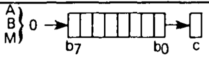
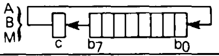
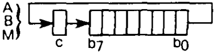
FIGURE 18 — PROGRAMMING AID

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0
		Immediate			Direct			Indexed			Extended			Inherent								
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	N	Z	V	C
ABX														3A	3	1	B + X→X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4+	2+	B9	5	3				A + M + C→A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4+	2+	F9	5	3				B + M + C→B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4+	2+	BB	5	3				A + M→A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4+	2+	FB	5	3				B + M→B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6+	2+	F3	7	3				D + M M + 1→D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4+	2+	B4	5	3				A Λ M→A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4+	2+	F4	5	3				B Λ M→B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC Λ IMM→CC					7
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6+	2+	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	•	↕
	ASRB													57	2	1		8	↕	↕	•	↕
	ASR				07	6	2	67	6+	2+	77	7	3					8	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4+	2+	B5	5	3				Bit Test A (M Λ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4+	2+	F5	5	3				Bit Test B (M Λ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0→A	•	0	1	0	0
	CLRB													5F	2	1	0→B	•	0	1	0	0
	CLR				0F	6	2	6F	6+	2+	7F	7	3				0→M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4+	2+	B1	5	3				Compare M from A	8	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4+	2+	F1	5	3				Compare M from B	8	↕	↕	↕	↕
	CMPD	10 83	5	4	10 93	7	3	10 A3	7+	3+	10 B3	8	4				Compare M M + 1 from D	•	↕	↕	↕	↕
	CMPS	11 8C	5	4	11 9C	7	3	11 AC	7+	3+	11 BC	8	4				Compare M M + 1 from S	•	↕	↕	↕	↕
	CMPU	11 83	5	4	11 93	7	3	11 A3	7+	3+	11 B3	8	4				Compare M M + 1 from U	•	↕	↕	↕	↕
	CMPX	8C	4	3	9C	6	2	AC	6+	2+	BC	7	3				Compare M M + 1 from X	•	↕	↕	↕	↕
	CMPY	10 8C	5	4	10 9C	7	3	10 AC	7+	3+	10 BC	8	4				Compare M M + 1 from Y	•	↕	↕	↕	↕
COM	COMA													43	2	1	$\overline{A}$→A	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B}$→B	•	↕	↕	0	1
	COM				03	6	2	63	6+	2+	73	7	3				$\overline{M}$→M	•	↕	↕	0	1
CWAI		3C	≥20	2													CC Λ IMM→CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1→A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1→B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6+	2+	7A	7	3				M − 1→M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4+	2+	B8	5	3				A ⊻ M→A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4+	2+	F8	5	3				B ⊻ M→B	•	↕	↕	0	•
EXG	R1, R2	1E	8	2													R1↔R2[2]	•	•	•	•	•
INC	INCA													4C	2	1	A + 1→A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1→B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6+	2+	7C	7	3				M + 1→M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3+	2+	7E	4	3				EA[3]→PC	•	•	•	•	•
JSR					9D	7	2	AD	7+	2+	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4+	2+	B6	5	3				M→A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4+	2+	F6	5	3				M→B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5+	2+	FC	6	3				M M + 1→D	•	↕	↕	0	•
	LDS	10 CE	4	4	10 DE	6	3	10 EE	6+	3+	10 FE	7	4				M M + 1→S	•	↕	↕	0	•
	LDU	CE	3	3	DE	5	2	EE	5+	2+	FE	6	3				M M + 1→U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5+	2+	BE	6	3				M M + 1→X	•	↕	↕	0	•
	LDY	10 8E	4	4	10 9E	6	3	10 AE	6+	3+	10 BE	7	4				M M + 1→Y	•	↕	↕	0	•
LEA	LEAS							32	4+	2+							EA[3]→S	•	•	•	•	•
	LEAU							33	4+	2+							EA[3]→U	•	•	•	•	•
	LEAX							30	4+	2+							EA[3]→X	•	•	↕	•	•
	LEAY							31	4+	2+							EA[3]→Y	•	•	↕	•	•

LEGEND:

OP	Operation Code (Hexadecimal)
~	Number of MPU Cycles
#	Number of Program Bytes
+	Arithmetic Plus
−	Arithmetic Minus
•	Multiply
$\overline{M}$	Complement of M
→	Transfer Into
H	Half-carry (from bit 3)
N	Negative (sign bit)
Z	Zero result
V	Overflow, 2's complement
C	Carry from ALU
↕	Test and set if true, cleared otherwise
•	Not Affected
CC	Condition Code Register
:	Concatenation
V	Logical or
Λ	Logical and
⊻	Logical Exclusive or

FIGURE 18 — PROGRAMMING AID (CONTINUED)

		Addressing Modes																5	3	2	1	0
		Immediate			Direct			Indexed[1]			Extended			Inherent								
Instruction	Forms	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Description	H	N	Z	V	C
LSL	LSLA													48	2	1		•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1		•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}$ + 1 → A	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}$ + 1 → B	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}$ + 1 → M	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A V M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B V M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC V IMM → CC				7	
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1		•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1		•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A − M − C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B − M − C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M M+1	•	↕	↕	0	•
	STS				10 DF	6	3	10 EF	6+	3+	10 FF	7	4				S → M M+1	•	↕	↕	0	•
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M M+1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M M+1	•	↕	↕	0	•
	STY				10 9F	6	3	10 AF	6+	3+	10 BF	7	4				Y → M M+1	•	↕	↕	0	•
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A − M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B − M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D − M M+1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI2[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI3[6]													11 3F	20	1	Software Interrupt 3	•	•	•	•	•
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2	1F	6	2													R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

NOTES:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET.

FIGURE 18 — PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~ 5	#	Description	5 H	3 N	2 Z	1 V	0 C
BCC	BCC	24	3	2	Branch C=0	•	•	•	•	•
	LBCC	10 24	5(6)	4	Long Branch C=0	•	•	•	•	•
BCS	BCS	25	3	2	Branch C=1	•	•	•	•	•
	LBCS	10 25	5(6)	4	Long Branch C=1	•	•	•	•	•
BEQ	BEQ	27	3	2	Branch Z=1	•	•	•	•	•
	LBEQ	10 27	5(6)	4	Long Branch Z=1	•	•	•	•	•
BGE	BGE	2C	3	2	Branch≥Zero	•	•	•	•	•
	LBGE	10 2C	5(6)	4	Long Branch≥Zero	•	•	•	•	•
BGT	BGT	2E	3	2	Branch>Zero	•	•	•	•	•
	LBGT	10 2E	5(6)	4	Long Branch>Zero	•	•	•	•	•
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10 22	5(6)	4	Long Branch Higher	•	•	•	•	•
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10 24	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
BLE	BLE	2F	3	2	Branch≤Zero	•	•	•	•	•
	LBLE	10 2F	5(6)	4	Long Branch≤Zero	•	•	•	•	•
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10 25	5(6)	4	Long Branch Lower	•	•	•	•	•

Instruction	Forms	Addressing Mode Relative OP	~ 5	#	Description	5 H	3 N	2 Z	1 V	0 C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10 23	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
BLT	BLT	2D	3	2	Branch<Zero	•	•	•	•	•
	LBLT	10 2D	5(6)	4	Long Branch<Zero	•	•	•	•	•
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10 2B	5(6)	4	Long Branch Minus	•	•	•	•	•
BNE	BNE	26	3	2	Branch Z=0	•	•	•	•	•
	LBNE	10 26	5(6)	4	Long Branch Z=0	•	•	•	•	•
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10 2A	5(6)	4	Long Branch Plus	•	•	•	•	•
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10 21	5	4	Long Branch Never	•	•	•	•	•
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V=0	•	•	•	•	•
	LBVC	10 28	5(6)	4	Long Branch V=0	•	•	•	•	•
BVS	BVS	29	3	2	Branch V=1	•	•	•	•	•
	LBVS	10 29	5(6)	4	Long Branch V=1	•	•	•	•	•

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N=1	BMI	2B	BPL	2A
Z=1	BEQ	27	BNE	26
V=1	BVS	29	BVC	28
C=1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r>m	BGT	2E	BLE	2F
r≥m	BGE	2C	BLT	2D
r=m	BEQ	27	BNE	26
r≤m	BLE	2F	BGT	2E
r<m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r>m	BHI	22	BLS	23
r≥m	BHS	24	BLO	25
r=m	BEQ	27	BNE	26
r≤m	BLS	23	BHI	22
r<m	BLO	25	BHS	24

NOTES:

1. All conditional branches have both short and long variations.
2. All short branches are 2 bytes and require 3 cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require 4 bytes and 6 cycles if the branch is taken or 5 cycles if the branch is not taken
5. 5(6) means. 5 cycles if branch not taken, 6 cycles if taken.

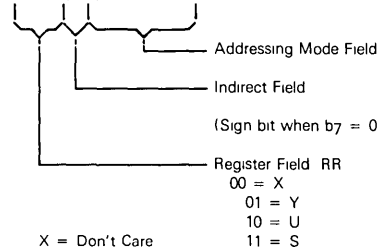
INDEXED ADDRESSING MODES

		Nondirect				Indirect			
Type	**Forms**	**Assembler Form**	**Post-Byte Opcode**	**+ ~**	**+ #**	**Assembler Form**	**Post-Byte Opcode**	**+ ~**	**+ #**
Constant Offset From R	No Offset	, R	1RR00100	0	0	[, R]	1RR10100	3	0
	5-Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8-Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16-Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R	A – Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B – Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D – Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	, R+	1RR00000	2	0	not allowed			
	Increment By 2	, R + +	1RR00001	3	0	[, R + +]	1RR10001	6	0
	Decrement By 1	, -R	1RR00010	2	0	not allowed			
	Decrement By 2	, --R	1RR00011	3	0	[, --R]	1RR10011	6	0
Constant Offset From PC	8-Bit Offset	n, PCR	1XX01100	1	1	[n, PCR]	1XX11100	4	1
	16-Bit Offset	n, PCR	1XX01101	5	2	[n, PCR]	1XX11101	8	2
Extended Indirect	16-Bit Address	–	–	–	–	[n]	10011111	5	2

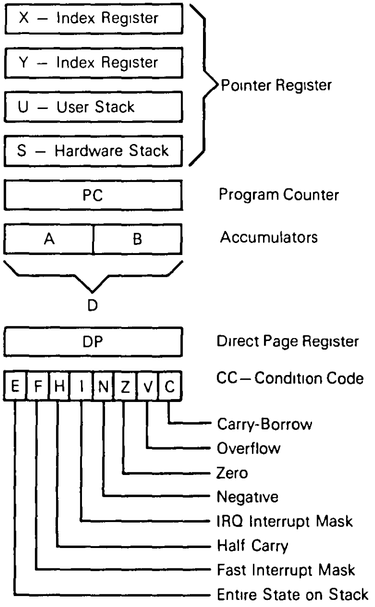
R = X, Y, U, or S RR 00 = X 10 = U
X = Don't Care 01 = Y 11 = S

INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

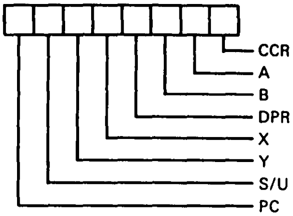
Post-Byte Register Bit								Indexed Addressing Mode
7	6	5	4	3	2	1	0	
0	R	R	x	x	x	x	x	EA = , R + 5 Bit Offset
1	R	R	0	0	0	0	0	, R +
1	R	R	I	0	0	0	1	, R + +
1	R	R	0	0	0	1	0	,- R
1	R	R	I	0	0	1	1	, -- R
1	R	R	I	0	1	0	0	EA = , R + 0 Offset
1	R	R	I	0	1	0	1	EA = , R + ACCB Offset
1	R	R	I	0	1	1	0	EA = , R + ACCA Offset
1	R	R	I	1	0	0	0	EA = , R+ 8-Bit Offset
1	R	R	I	1	0	0	1	EA = , R + 16-Bit Offset
1	R	R	I	1	0	1	1	EA = , R + D Offset
1	x	x	I	1	1	0	0	EA = , PC + 8-Bit Offset
1	x	x	I	1	1	0	1	EA = , PC + 16-Bit Offset
1	R	R	1	1	1	1	1	EA = [, Address]


Addressing Mode Field
Indirect Field
(Sign bit when b_7 = 0)
Register Field RR
00 = X
01 = Y
10 = U
11 = S
X = Don't Care

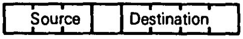
6809 PROGRAMMING MODEL


X – Index Register
Y – Index Register
U – User Stack
S – Hardware Stack
Pointer Register
PC
Program Counter
A
B
Accumulators
D
DP
Direct Page Register
E F H I N Z V C
CC – Condition Code
Carry-Borrow
Overflow
Zero
Negative
IRQ Interrupt Mask
Half Carry
Fast Interrupt Mask
Entire State on Stack

Push/Pull Post Byte



Transfer/Exchange Post Byte



Register Field

0000 = D (A-B)
0001 = X
0010 = Y
0011 = U
0100 = S
0101 = PC
1000 = A
1001 = B
1010 = CCR
1011 = DPR

6809 Stacking Order

Pull Order
↓
CC
A
B
DP
X Hi
X Lo
Y Hi
Y Lo
U/S Hi
U/S Lo
PC Hi
PC Lo
↑
Push Order
↓
Increasing Memory

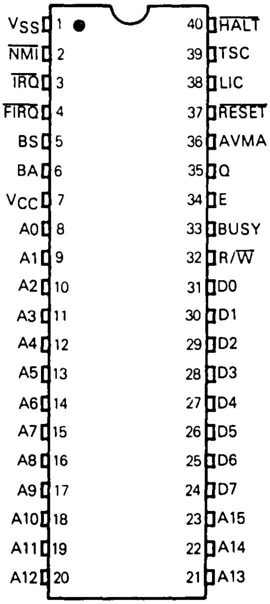
6809 Vectors

FFFE	Restart
FFFC	NMI
FFFA	SWI
FFF8	IRQ
FFF6	FIRQ
FFF4	SW12
FFF2	SW13
FFF0	Reserved

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6809EP
	1.0 MHz	−40°C to 85°C	MC6809ECP
	1.5 MHz	0°C to 70°C	MC68A09EP
	1.5 MHz	−40°C to 85°C	MC68A09ECP
	2.0 MHz	0°C to 70°C	MC68B09EP
	2.0 MHz	−40°C to 85°C	MC68B09ECP
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6809ES
	1.0 MHz	−40°C to 85°C	MC6809ECS
	1.5 MHz	0°C to 70°C	MC68A09ES
	1.5 MHz	−40°C to 85°C	MC68A09ECS
	2.0 MHz	0°C to 70°C	MC68B09ES
	2.0 MHz	−40°C to 85°C	MC68B09CS

PIN ASSIGNMENT


VSS 1
NMI 2
IRQ 3
FIRQ 4
BS 5
BA 6
VCC 7
A0 8
A1 9
A2 10
A3 11
A4 12
A5 13
A6 14
A7 15
A8 16
A9 17
A10 18
A11 19
A12 20
40 HALT
39 TSC
38 LIC
37 RESET
36 AVMA
35 Q
34 E
33 BUSY
32 R/W
31 D0
30 D1
29 D2
28 D3
27 D4
26 D5
25 D6
24 D7
23 A15
22 A14
21 A13

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MCM6810

128 × 8-Bit Random-Access Memory

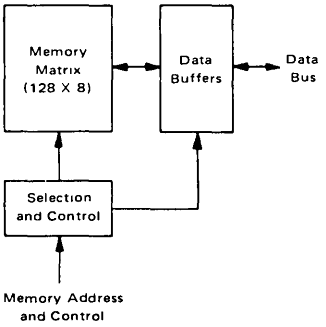
The MCM6810 is a byte-organized memory designed for use in bus-organized systems. It is fabricated with N-channel silicon-gate technology. For ease of use, the device operates from a single power supply, has compatibility with TTL and DTL, and needs no clocks or refreshing because of static operation.

The memory is compatible with the M6800 Microcontroller Family, providing random storage in byte increments. Memory expansion is provided through multiple Chip Select inputs.

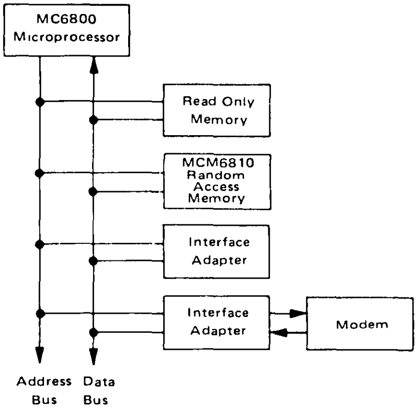
- Organized as 128 Bytes of 8 Bits
- Static Operation
- Bidirectional Three-State Data Input/Output
- Six Chip Select Inputs (Four Active Low, Two Active High)
- Single 5-Vol Power Supply
- TTL Compatible
- Maximum Access Time = 450 ns — MCM6810
  360 ns — MCM68A10
  250 ns — MCM68B10

This document contains information on a new product. Specifications and information herein are subject to change without notice:

MCM6810 RANDOM ACCESS MEMORY BLOCK DIAGRAM


Memory Matrix (128 X 8)
Data Buffers
Data Bus
Selection and Control
Memory Address and Control

M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM


MC6800 Microprocessor
Read Only Memory
MCM6810 Random Access Memory
Interface Adapter
Interface Adapter
Modem
Address Bus
Data Bus

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MCM6810, MCM68A10, MCM68B10 MCM6810C, MCM68A10C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 120 65	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \tag{1}$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

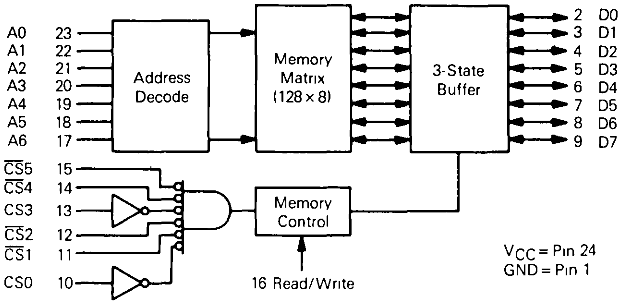
An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \tag{2}$$

Solving equations (1) and (2) for K gives $K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

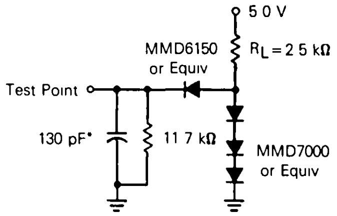
BLOCK DIAGRAM


A0 23
A1 22
A2 21
A3 20
A4 19
A5 18
A6 17
$\overline{CS5}$ 15
$\overline{CS4}$ 14
CS3 13
$\overline{CS2}$ 12
$\overline{CS1}$ 11
CS0 10
Address Decode
Memory Matrix (128 × 8)
3-State Buffer
Memory Control
16 Read/Write
2 D0
3 D1
4 D2
5 D3
6 D4
7 D5
8 D6
9 D7
V_{CC} = Pin 24
GND = Pin 1

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	V_{SS} + 2.0	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS} − 0.3	V_{SS} + 0.8	V
Input Current (A_n, R/$\overline{W}$, $\overline{CS}_n$) (V_{in} = 0 to 5.25 V)	I_{in}	–	2.5	µA
Output High Voltage (I_{OH} = −205 µA)	V_{OH}	2.4	–	V
Output Low Voltage (I_{OL} = 1.6 mA)	V_{OL}	–	0.4	V
Output Leakage Current (Three-State) (CS = 0.8 V or $\overline{CS}$ = 2.0 V, V_{out} = 0.4 V to 2.4 V)	I_{TSI}	–	10	µA
Supply Current 1.0 MHz (V_{CC} = 5.25 V, All Other Pins Grounded) 1.5, 2.0 MHz	I_{CC}	– –	80 100	mA
Input Capacitance (A_n, R/$\overline{W}$, CS_n, $\overline{CS}_n$) (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	7.5	pF
Output Capacitance (D_n) (V_{out} = 0, T_A = 25°C, f = 1.0 MHz, CS0 = 0)	C_{out}	–	12.5	pF

AC TEST LOAD


5.0 V
MMD6150 or Equiv
R_L = 2.5 kΩ
Test Point
130 pF*
11.7 kΩ
MMD7000 or Equiv

*Includes Jig Capacitance

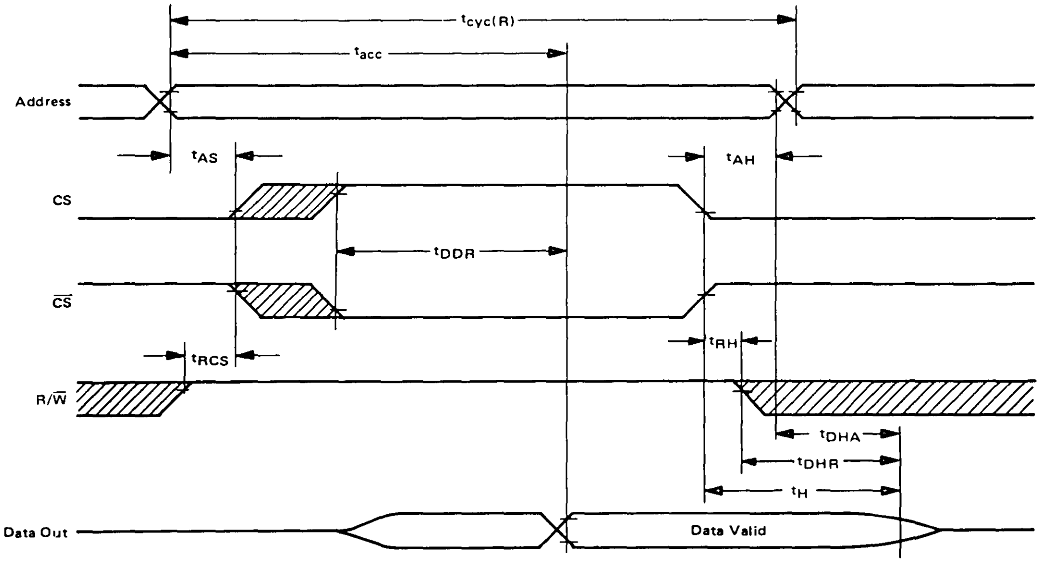
AC OPERATING CONDITIONS AND CHARACTERISTICS

READ CYCLE (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic	Symbol	MCM6810		MCM68A10		MCM68B10		Unit
		Min	Max	Min	Max	Min	Max	
Read Cycle Time	$t_{cyc(R)}$	450	–	360	–	250	–	ns
Access Time	t_{acc}	–	450	–	360	–	250	ns
Address Setup Time	t_{AS}	20	–	20	–	20	–	ns
Address Hold Time	t_{AH}	0	–	0	–	0	–	ns
Data Delay Time (Read)	t_{DDR}	–	230	–	220	–	180	ns
Read to Select Delay Time	t_{RCS}	0	–	0	–	0	–	ns
Data Hold from Address	t_{DHA}	10	–	10	–	10	–	ns
Output Hold Time	t_H	10	–	10	-	10	–	ns
Data Hold from Read	t_{DHR}	10	80	10	60	10	60	ns
Read Hold from Chip Select	t_{RH}	0	–	0	–	0	–	ns

3

READ CYCLE TIMING



NOTES
1 Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified
3 CS and $\overline{CS}$ have same timing

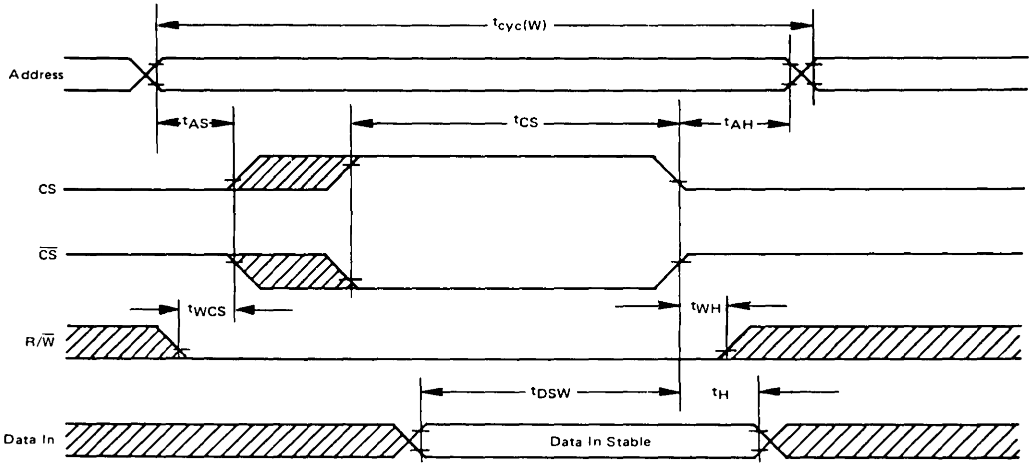
▨ = Don't Care

WRITE CYCLE (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic	Symbol	MCM6810		MCM68A10		MCM68B10		Unit
		Min	Max	Min	Max	Min	Max	
Write Cycle Time	$t_{cyc(W)}$	450	–	360	–	250	–	ns
Address Setup Time	t_{AS}	20	–	20	–	20	–	ns
Address Hold Time	t_{AH}	0	–	0	–	0	–	ns
Chip Select Pulse Width	t_{CS}	300	–	250	–	210	–	ns
Write to Chip Select Delay Time	t_{WCS}	0	–	0	–	0	–	ns
Data Setup Time (Write)	t_{DSW}	190	–	80	–	60	–	ns
Input Hold Time	t_H	10	–	10	–	10	–	ns
Write Hold Time from Chip Select	t_{WH}	0	–	0	–	0	–	ns

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not authorized for use as components in life support devices or systems intended for surgical implant into the body or intended to support or sustain life. Buyer agrees to notify Motorola of any such intended end use whereupon Motorola shall determine availability and suitability of its product or products for the use intended. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Employment Opportunity/Affirmative Action Employer.

WRITE CYCLE TIMING



$t_{cyc(W)}$
Address
t_{AS}
t_{CS}
t_{AH}
CS
$\overline{CS}$
t_{WCS}
t_{WH}
R/$\overline{W}$
t_{DSW}
t_H
Data In
Data In Stable

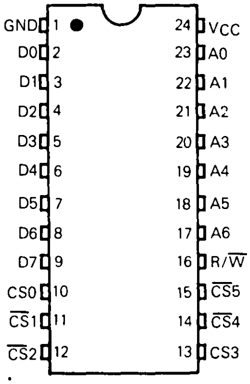
NOTES
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified
3. CS and $\overline{CS}$ have same timing

▨ = Don't Care

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Plastic P Suffix	1.0	0°C to 70°C	MCM6810P
	1.0	−40°C to 85°C	MCM6810CP
	1.5	0°C to 70°C	MCM68A10P
	1.5	−40°C to 85°C	MCM68A10CP
	2.0	0°C to 70°C	MCM68B10P
Cerdip S Suffix	1.0	0°C to 70°C	MCM6810S
	1.0	−40°C to 85°C	MCM6810CS
	1.5	0°C to 70°C	MCM68A10S
	1.5	−40°C to 85°C	MCM68A10CS
	2.0	0°C to 70°C	MCM68B10S

PIN ASSIGNMENTS



GND 1
D0 2
D1 3
D2 4
D3 5
D4 6
D5 7
D6 8
D7 9
CS0 10
$\overline{CS1}$ 11
$\overline{CS2}$ 12
24 V_{CC}
23 A0
22 A1
21 A2
20 A3
19 A4
18 A5
17 A6
16 R/$\overline{W}$
15 $\overline{CS5}$
14 $\overline{CS4}$
13 CS3

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11A0

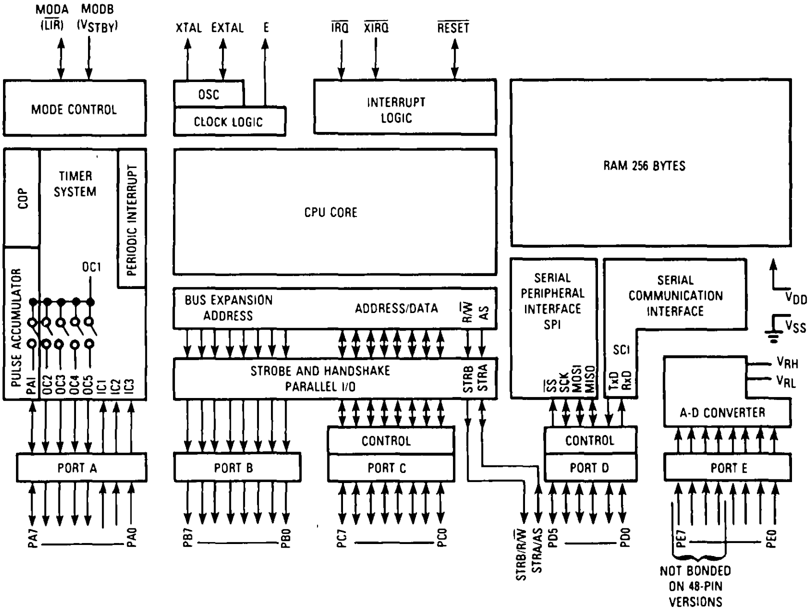
Technical Summary
8-Bit Microcontroller Unit

The MC68HC11A0 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 256 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

BLOCK DIAGRAM



MODA ($\overline{LIR}$)
MODB (V_{STBY})
XTAL
EXTAL
E
$\overline{IRQ}$
$\overline{XIRQ}$
$\overline{RESET}$
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
RAM 256 BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
PULSE ACCUMULATOR
OC1
PAI
OC2
OC3
OC4
OC5
IC1
IC2
IC3
CPU CORE
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/$\overline{W}$
AS
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SERIAL PERIPHERAL INTERFACE SPI
$\overline{SS}$
SCK
MOSI
MISO
SERIAL COMMUNICATION INTERFACE
SCI
TxD
RxD
V_{DD}
V_{SS}
V_{RH}
V_{RL}
A-D CONVERTER
CONTROL
PORT A
PORT B
CONTROL
PORT C
CONTROL
PORT D
PORT E
PA7 — PA0
PB7 — PB0
PC7 — PC0
STRB/R/$\overline{W}$
STRA/AS
PD5 — PD0
PE7 — PE0
NOT BONDED ON 48-PIN VERSIONS

This document contains information on a new product Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU. This mode would not normally be used on the MC68HC11A0, because of no internal ROM.

EXPANDED MULTIPLEXED MODE

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{\text{W}}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits. This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{\text{RESET}}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins.

$\overline{\text{IRQ}}$

This pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive or level-sensitive triggering is program selectable. This pin is configured to level sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{\text{IRQ}}$.

$\overline{\text{XIRQ}}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register is set, and any interrupt is masked until enabled by software. This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{\text{LIR}}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{\text{LIR}}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The mode selections are shown below.

MODB	MODA	MODE SELECTED
1	0	Single Chip*
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

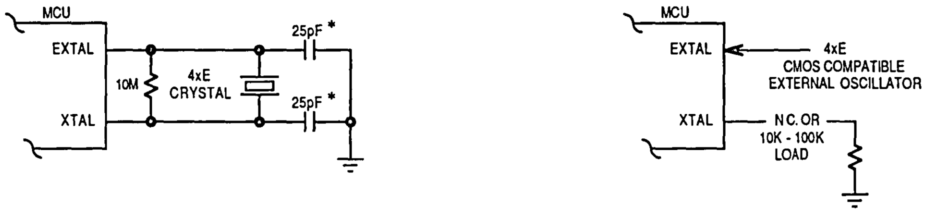
*This mode not useable due to no internal ROM.

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

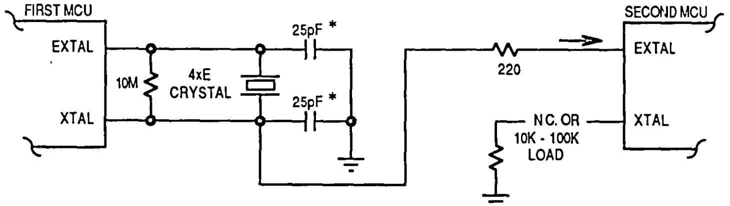
R/$\overline{\text{W}}$/STRB

This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides R/$\overline{\text{W}}$ (read-write) function. The R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.



Common Crystal Connections

External Oscillator Connections



*This value includes all stray capacitances.

One Crystal Driving Two MCUs

Figure 1. Oscillator Collections

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for three input capture functions; four output compare functions; and pulse accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of parallel I/O where the STRA input and STRB output acts

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/$\overline{\text{SCK}}$	PD4/$\overline{\text{SCK}}$
D-5	PD5/$\overline{\text{SS}}$	PD5$\overline{\text{SS}}$
	STRA	AS
	STRB	R/$\overline{\text{W}}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4##	PE4/AN4##
E-5	PE5/AN5##	PE5/AN5##
E-6	PE6/AN6##	PE6/AN6##
E-7	PE7/AN7##	PE7/AN7##

##Not Bonded in 48-Pin Versions

as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/$\overline{\text{W}}$ signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between the shaded areas (EXT) are for externally addressed memory and I/O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR A AND B

These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.

7	A	0	7	B	0
15		D			0

INDEX REGISTER X (IX)

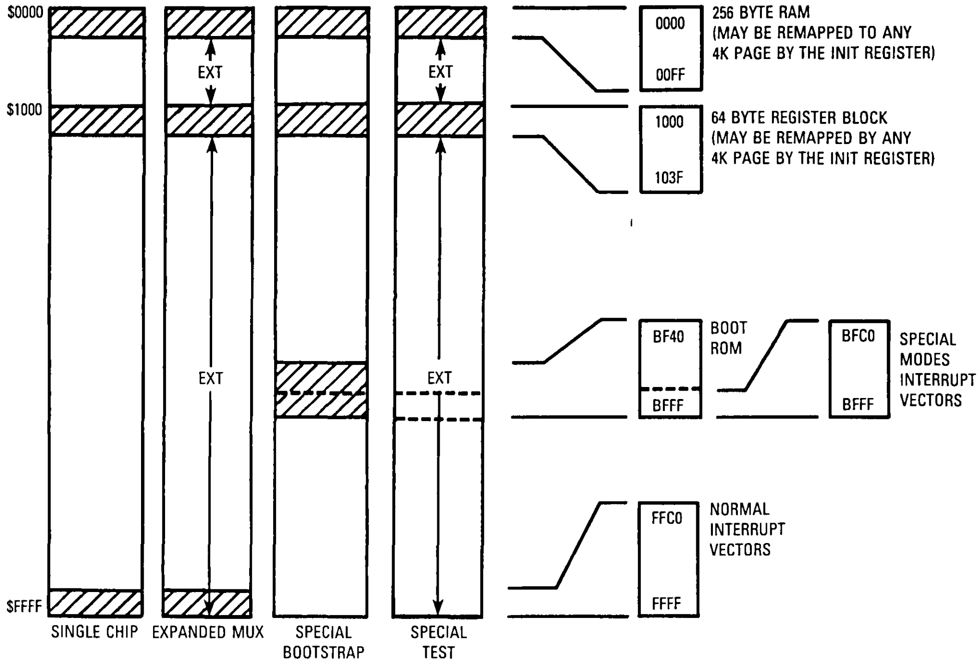
This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.

15	IX	0

INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.

15	IY	0


$0000
$1000
$FFFF
EXT
EXT
EXT
EXT
SINGLE CHIP
EXPANDED MUX
SPECIAL BOOTSTRAP
SPECIAL TEST
0000
00FF
256 BYTE RAM (MAY BE REMAPPED TO ANY 4K PAGE BY THE INIT REGISTER)
1000
103F
64 BYTE REGISTER BLOCK (MAY BE REMAPPED BY ANY 4K PAGE BY THE INIT REGISTER)
BF40
BFFF
BOOT ROM
BFC0
BFFF
SPECIAL MODES INTERRUPT VECTORS
FFC0
FFFF
NORMAL INTERRUPT VECTORS

NOTE:
*Either or both the internal RAM and registers can be remapped to any 4K boundary by software.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TI405	Output Compare 5 Register/ Input Capture 4 Register
$101F	Bit 7	—	—	—	—	—	—	Bit 0		
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	I405I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Reg. 1
$1023	OC1F	OC2F	OC3F	OC4F	I405F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Reg 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Reg 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Reg. 2
$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/05	RTR1	RTR0	PACTL	Pulse Accum Control Reg
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accum Count Reg.
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control

Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035				PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT	EEPROM Block Protect Reg.
$1036 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Circuitry
$103B	—	—	—	—	—	—	—	—	PPROG	EEPROM Prog.Control Reg.
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Reg.
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F					NOSEC	NOCOP	ROMON	EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

15	PC	0

STACK POINTER (SP)

The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the accumulators A and B and registers IX and IY can be stored during certain instructions.

15	SP	0

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

7							0
S	X	H	I	N	Z	V	C

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or $\overline{\text{XIRQ}}$) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

RESETS

The MCU can be reset four ways: 1) an active low input to the $\overline{\text{RESET}}$ pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

RESET PIN

To request an external reset, the $\overline{\text{RESET}}$ pin must be held low for eight E_{cyc} (two E_{cyc} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external $\overline{\text{RESET}}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system.

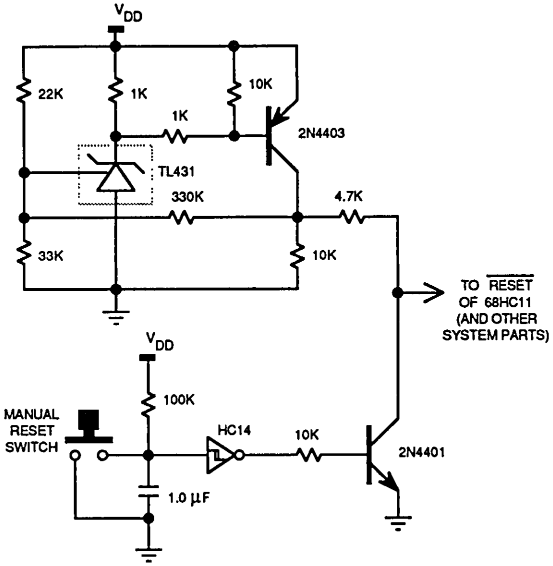
The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

CLOCK MONITOR RESET

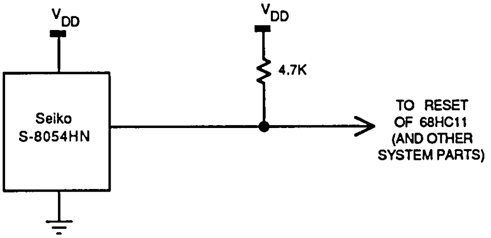
The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the $\overline{\text{RESET}}$ pin is driven low to reset the external system.

Table 2. COP Timeout Periods

CR1	CR0	$E/2^{15}$ Divided By	XTAL = 2^{23} Timeout −1/+15.6 ms	XTAL = 8.0 MHz Timeout −0/+16.4 ms	XTAL = 4.9152 MHz Timeout −0/+26.7 ms	XTAL = 4.0 MHz Timeout −0/+32.8 ms	XTAL = 3.6864 MHz Timeout −0/+35.6 ms
0	0	1	15.625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz


VDD
22K
1K
10K
1K
2N4403
TL431
330K
4.7K
33K
10K
TO $\overline{\text{RESET}}$
OF 68HC11
(AND OTHER
SYSTEM PARTS)
VDD
100K
MANUAL
RESET
SWITCH
HC14
10K
2N4401
1.0 µF

Reset Circuit with LVI and RC Delay


VDD
Seiko
S-8054HN
VDD
4.7K
TO RESET
OF 68HC11
(AND OTHER
SYSTEM PARTS)

Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.

INTERRUPTS

There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the $\overline{\text{XIRQ}}$ pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the $\overline{\text{XIRQ}}$ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
*	*		
*	*		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Output Compare 5	I Bit	OC5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	IRQ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	XIRQ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	RESET	None	None

interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

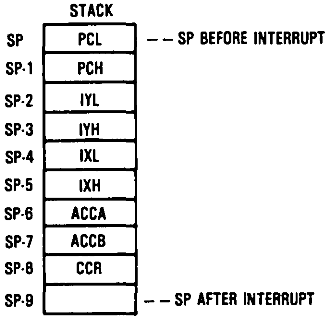


Figure 4. Stacking Order

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be E/2^{13}, E/2^{14}, E/2^{15}, or E/2^{16}.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either IRQ, XIRQ or RESET. An external interrupt used at IRQ is only efective if the I bit in the CCR is clear. An external interrupt applied at the XIRQ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the XIRQ request. If the X bit is set, the processing will continue (if no XIRQ interrupt service routine is requested) with the instruction immediately following the STOP instruction. A low input to the RESET pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

If the internal oscillator is being used, a restart delay is required to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the RESET pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit, and the restart delay will be imposed.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes slightly more power than the STOP mode. In the WAIT mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or RESET. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the WAIT mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces WAIT-mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from $FFFF to $0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has three input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are three 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

TIMER CONTROL REGISTER 2 (TCTL2)

	7	6	5	4	3	2	1	0
	0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET	0	0	0	0	0	0	0	0

Bits 7-6 — Not Implemented
These bits always read zero.

EDGxB and EDGxA — Input Capture x Edge Control
These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to $FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one.

TIMER COMPARE FORCE REGISTER (CFORC)

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

7	6	5	4	3	2	1	0
FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET							
0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action
1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set
0 = Has no meaning
Bits 2-0 — Not Implemented
These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M)

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

7	6	5	4	3	2	1	0
OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET							
0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D)

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

7	6	5	4	3	2	1	0
OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET							
0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1)

7	6	5	4	3	2	1	0
OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET							
0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode
OL2-OL5 — Output Level
These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TIMER INTERRUPT MASK REGISTER 1 (TMSK1)

7	6	5	4	3	2	1	0
OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
RESET							
0	0	0	0	0	0	0	0

OCxI — Output Compare x Interrupt
1 = Interrupt sequence requested if OCxF = 1 in TFLG1
0 = Interrupt inhibited
ICxI — Input Capture x Interrupt
1 = Interrupt sequence requested if ICxF = 1 in TFLG1
0 = Interrupt inhibited

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position.

7	6	5	4	3	2	1	0
OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F
RESET							
0	0	0	0	0	0	0	0

OCxF — Output Compare x Flag
Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected
ICxF — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected

TIMER INTERRUPT MASK REGISTER 2 (TMSK2)

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG2. Two timer prescaler bits are also included in this register.

7	6	5	4	3	2	1	0
TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET							
0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled
RTII — RTI Interrupt Enable
1 = Interrupt requested when RTIF = 1
0 = RTIF interrupt disabled
PAOVI — Pulse Accumulator Overflow Interrupt Enable
1 = Interrupt requested when PAOVF = 1
0 = PAOVF disabled

PAII — Pulse Accumulator Input Interrupt Enable
1 = Interrupt requested when PAIF = 1
0 = PAIF disabled

Bits 3-2 — Not Implemented
These bits always read zero.

PR1 and PR0 — Timer Prescaler Selects
Can only be written to during initialization. Writes are disabled after the first write or after 64 E cycles out of reset.

PR1	PR0	Divide-by-Factor
0	0	1
0	1	4
1	0	8
1	1	16

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2)

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

	7	6	5	4	3	2	1	0
	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

TOF — Timer Overflow
Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag
Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag
Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag
Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented
These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0
	DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0
RESET	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
1 = Output
0 = Input only

PAEN — Pulse-Accumulator System Enable
1 = Pulse accumulator on
0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode
1 = Gated time accumulator
0 = External even counting

PEDGE — Pulse Accumulator Edge Control
This bit provides clock action along with PAMOD.
1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.
0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

Bits 3-2 — Not Implemented
These bits always read zero.

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

SYSTEM CONFIGURATION REGISTER (CONFIG)

The CONFIG register comes set as $0C. Changing bits 0 or 1 could level to accessing an undefined ROM and a defective EEPROM.

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

7	6	5	4	3	2	1	0
0	0	0	0	0	NOCOP	ROMON	EEON

Bits 7-3 - Not Implemented
These bits are always read as zero.

NOCOP — COP System Disable
1 = COP watchdog system disabled
0 = COP watchdog system enabled

ROMON — Enable On-Chip ROM
This bit is programmed to "zero" at manufacturing, disabling the 8K ROM. The memory space becomes externally accessed space.

EEON — Enable On-Chip EEPROM
This bit is programmed to "zero" at manufacturing, disabling the 512-byte EEPROM. The memory space becomes externally accessed space.

SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins: PD0 for receive data (RxD) and PD1 for the transmit data (TxD). The baud-rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/reception of a message;
2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR)

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1)

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

	7	6	5	4	3	2	1	0
	R8	T8	0	M	WAKE	0	0	0
RESET	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.

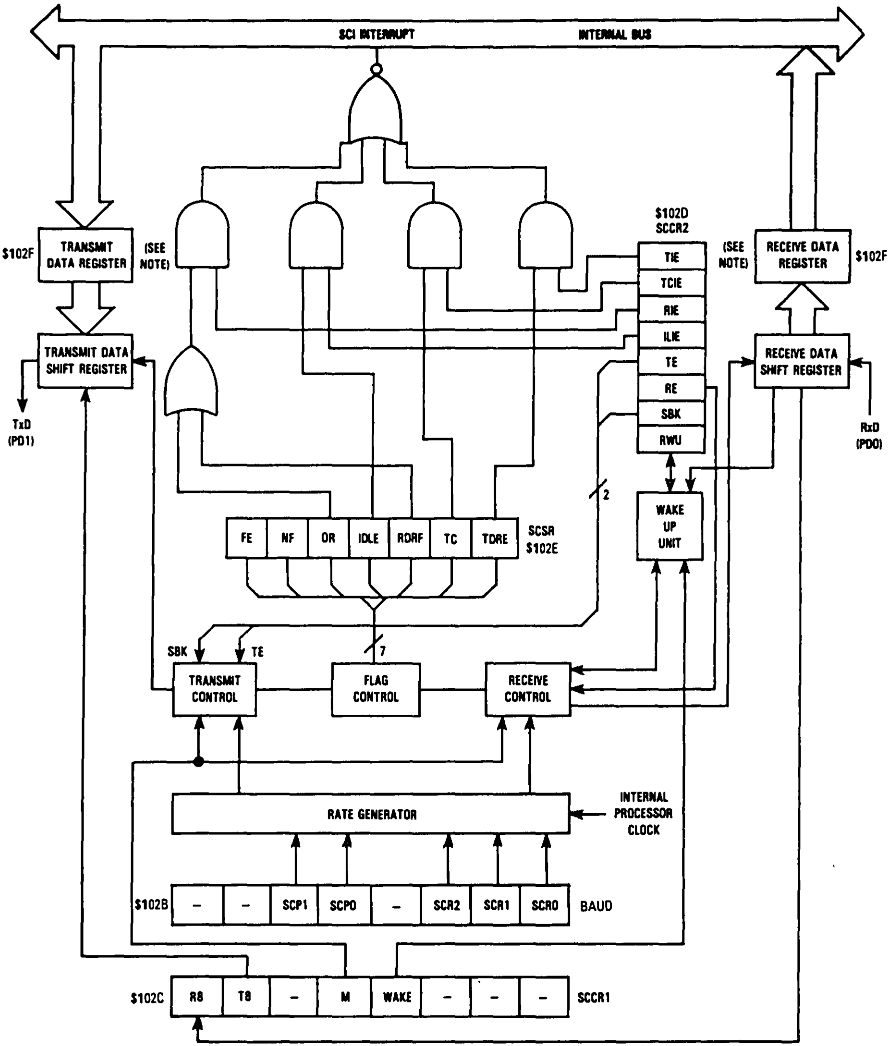
T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.

Bit 5 — Not Implemented
This bit always reads zero.

M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line

Bits 2-0 — Not Implemented
These bits always read zero.


SCI INTERRUPT
INTERNAL BUS
$102F
TRANSMIT DATA REGISTER
(SEE NOTE)
TRANSMIT DATA SHIFT REGISTER
TxD (PD1)
$102D SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(SEE NOTE)
RECEIVE DATA REGISTER
$102F
RECEIVE DATA SHIFT REGISTER
RxD (PD0)
2
WAKE UP UNIT
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $102E
7
SBK
TE
TRANSMIT CONTROL
FLAG CONTROL
RECEIVE CONTROL
RATE GENERATOR
INTERNAL PROCESSOR CLOCK
$102B
SCP1
SCP0
SCR2
SCR1
SCR0
BAUD
$102C
R8
T8
M
WAKE
SCCR1

NOTE: The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read.

Figure 5. SCI Block Diagram

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt if TDRE = 1
0 = TDR interrupts disabled

TCIE — Transmit-Complete Interrupt Enable
1 = SCI interrupt if TC = 1
0 = TC interrupts disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt if RDRF or OR = 1
0 = RDRF or OR interrupt disabled

ILIE — Idle-Line Interrupt Enable
1 = SCI interrupt if IDLE = 1
0 = IDLE interrupts disabled

TE — Transmit Enable
1 = Transmit shift register output is applied to the TxD line
0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
1 = Receiver enabled
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited

RWU — Receiver Wake Up
When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the WAKE bit is zero, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break
If this bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET	1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty
1 = Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register
0 = Cleared by a read of SCSR (with TDRE = 1) followed by a write to SCDR

TC — Transmit Complete
1 = Automatically set when all data frame, preamble, or break condition transmissions are complete
0 = Cleared by a read of SCSR (with TC = 1) followed by a write to SCDR

RDRF — Receive Data Register Full
1 = Automatically set when a character is transferred from the receiver shift register to the SCDR
0 = Cleared by a read of SCSR (with RDRF = 1) followed by a read of SCDR

IDLE — Idle-Line Detect
This bit is inhibited while RWU = 1.
1 = Automatically set when the receiver serial input becomes idle after having been active
0 = Cleared by a read of SCSR (with IDLE = 1) followed by a read of SCDR

OR — Overrun Error
1 = Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read
0 = Cleared by a read of SCSR (with OR = 1) followed by a read of SCDR

NF — Noise Flag
1 = Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame
0 = Cleared by a read of SCSR (with NF = 1) followed by a write to SCDR

FE — Framing Error
1 = Automatically set when a logic 0 is detected where a stop bit was expected
0 = Cleared by a read of SCSR (with FE = 1) followed by a read of SCDR

Bit 0 — Not Implemented
This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

	7	6	5	4	3	2	1	0
	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET	0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)
This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.

Bit 6 — Not Implemented
This bit always reads zero.

SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects
These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)
This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

SCR2-SCR0 — SCI Baud-Rate Selects
These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	1	0		131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt if SPIF = 1
0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

DWOM — Port D Wire-OR Mode Option
This bit affects all six port D pins together.
1 = Port D outputs act as open-drain outputs
0 = Port D outputs are normal CMOS outputs

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
This bit selects the polarity of the SCK clock.
1 = SCK line idles high
0 = SCK line idles low

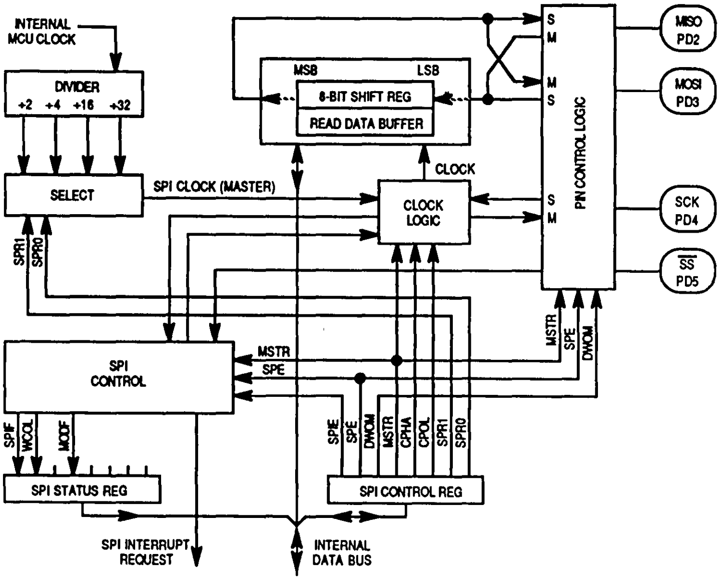

INTERNAL MCU CLOCK
DIVIDER
÷2 ÷4 ÷16 ÷32
SELECT
SPI CLOCK (MASTER)
SPR1
SPR0
MSB
LSB
8-BIT SHIFT REG
READ DATA BUFFER
CLOCK
CLOCK LOGIC
S
M
M
S
S
M
PIN CONTROL LOGIC
MISO PD2
MOSI PD3
SCK PD4
SS PD5
MSTR
SPE
DWOM
SPI CONTROL
MSTR
SPE
SPIF
WCOL
MODF
SPI STATUS REG
SPIE
SPE
DWOM
MSTR
CPHA
CPOL
SPR1
SPR0
SPI CONTROL REG
SPI INTERRUPT REQUEST
INTERNAL DATA BUS

Figure 6. SPI Block Diagram

CPHA — Clock Phase

This bit selects one of two fundamentally different clock protocols. Refer to Figure 7.

SPR1 and SPR0 — SPI Clock Rate Select

These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR)

	7	6	5	4	3	2	1	0
	SPIF	WCOL	0	MODF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag

1 = Automatically set when data transfer is complete between processor and external device

0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision

If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{SS}$ is low and ends when the SPIF flag gets set.

1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred

0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented

This bit always reads zero.

MODF — Mode Fault

This bit indicates the possibility of a multi-master conflict for system control and therefore allows a proper exit from system operation to a reset or default system state.

1 = Automatically set when a master device has its $\overline{SS}$ pin pulled low

0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented

These bits always read zero.

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.
2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

NOTE

In the 48-pin dual-in-line package, four conversion channels are not implemented. These include channels four through seven.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

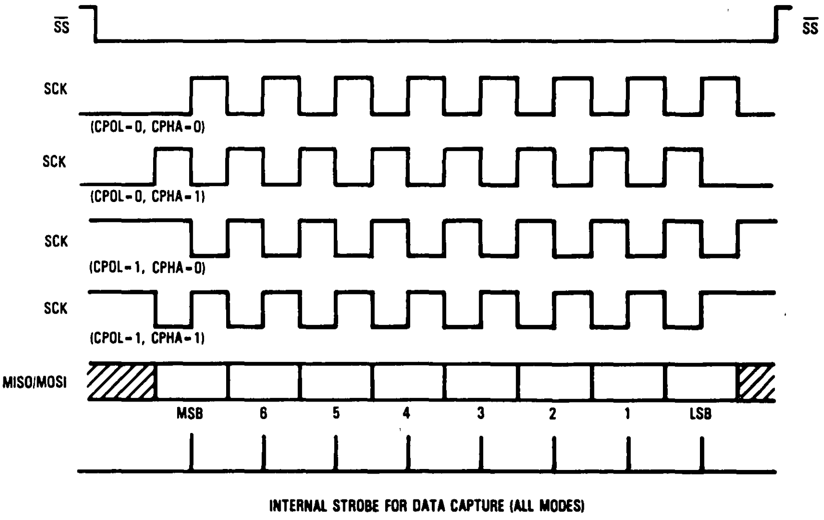


Figure 7. Data Clock Timing Diagram

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16×16	FDIV
Integer Divide 16×16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8×8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)

— Continued —

Function	Mnemonic
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

— Continued —

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if = zero	BEQ
Branch if≥zero	BGE
Branch if)zero	BGT
Branch if Higher	BHI
Branch if≤Zero	BLE
Branch if Lower or Same	BLS
Branch if(Zero	BLT
Branch if Minus	BMI
Branch if not = Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN

— Continued —

Function	Mnemonic
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

OPCODE MAP SUMMARY

Table 6 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

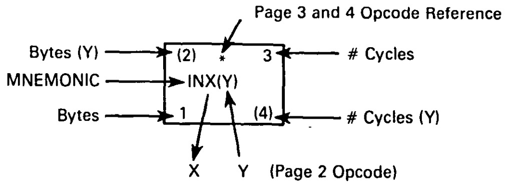
Table 6. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
HI / LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LOW
0 0000	TEST 1	SBA 2 1	BRA 3 2	(2) TSX(Y) 3 1 (4)	NEGA 2 1	NEGB 2 1	(3) NEG 6 2 (7)	NEG 6 3	SUBA 2 2	SUBA 3 2	(3) SUBA 4 2 (5)	SUBA 4 3	SUBB 2 2	SUBB 3 2	(3) SUBB 4 2 (5)	SUBB 4 3	0
1 0001	NOP 2 1	CBA 2 1	BRN 3 2	INS 3 1					CMPA 2 2	CMPA 3 2	(3) CMPA 4 2 (5)	CMPA 4 3	CMPB 2 2	CMPB 3 2	(3) CMPB 4 2 (5)	CMPB 4 3	1
2 0010	IDIV 41 1	BRSET 6 4	BHI 3 2	PULA 4 1					SBCA 2 2	SBCA 3 2	(3) SBCA 4 2 (5)	SBCA 4 3	SBCB 2 2	SBCB 3 2	(3) SBCB 4 2 (5)	SBCB 4 3	2
3 0011	FDIV 41 1	BRCLR 6 4	BLS 3 2	PULB 4 1	COMA 2 1	COMB 2 1	(3) COM 6 2 (7)	COM 6 3	* 4 3 SUBD	* 5 2 SUBD	(3) * 6 2 SUBD (7)	* 6 3 SUBD	ADDD 4 3	ADDD 5 2	(3) ADDD 6 2 (7)	ADDD 6 3	3
4 0100	LSRD 3 1	BSET 6 3	(BHS) BCC 3 2	DES 3 1	LSRA 2 1	LSRB 2 1	(3) LSR 6 2 (7)	LSR 6 3	ANDA 2 2	ANDA 3 2	(3) ANDA 4 2 (5)	ANDA 4 3	ANDB 2 2	ANDB 3 2	(3) ANDB 4 2 (5)	ANDB 4 3	4
5 0101	(LSLD) ASLD 3 1	BCLR 6 3	(BLO) BCS 3 2	(2) TX(Y)S 3 1 (4)					BITA 2 2	BITA 3 2	(3) BITA 4 2 (5)	BITA 4 3	BITB 2 2	BITB 3 2	(3) BITB 4 2 (5)	BITB 4 3	5
6 0110	TAP 2 1	TAB 2 1	BNE 3 2	PSHA 3 1	RORA 2 1	RORB 2 1	(3) ROR 6 2 (7)	ROR 6 3	LDAA 2 2	LDAA 3 2	(3) LDAA 4 2 (5)	LDAA 4 3	LDBB 2 2	LDBB 3 2	(3) LDBB 4 2 (5)	LDBB 4 3	6
7 0111	TPA 2 1	TBA 2 1	BEQ 3 2	PSHB 3 1	ASRA 2 1	ASRB 2 1	(3) ASR 6 2 (7)	ASR 6 3		STAA 3 2	(3) STAA 4 2 (5)	STAA 4 3		STBB 3 2	(3) STBB 4 2 (5)	STBB 4 3	7
8 1000	(2) INX(Y) 3 1 (4)	PAGE 2	BVC 3 2	(2) PULX(Y) 5 1 (6)	ASLA 2 1	ASLB 2 1	(3) ASL 6 2 (7)	ASL 6 3	EORA 2 2	EORA 3 2	(3) EORA 4 2 (5)	EORA 4 3	EORB 2 2	EORB 3 2	(3) EORB 4 2 (5)	EORB 4 3	8
9 1001	(2) DEX(Y) 3 1 (4)	DAA 2 1	BVS 3 2	RTS 5 1	ROLA 2 1	ROLB 2 1	(3) ROL 6 2 (7)	ROL 6 3	ADCA 2 2	ADCA 3 2	(3) ADCA 4 2 (5)	ADCA 4 3	ADCB 2 2	ADCB 3 2	(3) ADCB 4 2 (5)	ADCB 4 3	9
A 1010	CLV 2 1	PAGE 3	BPL 3 2	(2) ABX(Y) 3 1 (4)	DECA 2 1	DECB 2 1	(3) DEC 6 2 (7)	DEC 6 3	ORAA 2 2	ORAA 3 2	(3) ORAA 4 2 (5)	ORAA 4 3	ORAB 2 2	ORAB 3 2	(3) ORAB 4 2 (5)	ORAB 4 3	A
B 1011	SEV 2 1	ABA 2 1	BMI 3 2	RTI 12 1					ADDA 2 2	ADDA 3 2	(3) ADDA 4 2 (5)	ADDA 4 3	ADDB 2 2	ADDB 3 2	(3) ADDB 4 2 (5)	ADDB 4 3	B
C 1100	CLC 2 1	(4) BSET 7 3 (8)	BGE 3 2	(2) PSHX(Y) 4 1 (5)	INCA 2 1	INCB 2 1	(3) INC 6 2 (7)	INC 6 3	(4) CPX(Y) 4 3 5	(3) CPX(Y) 5 2 (6)	(3) * 6 2 CPX(Y) (7)	(4) CPX(Y) 6 3 (7)	LDD 3 3	LDD 4 2	(3) LDD 5 2 (6)	LDD 5 3	C
D 1101	SEC 2 1	(4) BCLR 7 3 (8)	BLT 3 2	MUL 10 1	TSTA 2 1	TSTB 2 1	(3) TST 6 2 (7)	TST 6 3	BSR 6 2	JSR 5 2	(3) JSR 6 2 (7)	JSR 6 3	PAGE 4	STD 4 2	(3) STD 5 2 (6)	STD 5 3	D
E 1110	CLI 2 1	(5) BRSET 7 4 (8)	BGT 3 2	WAI 12 1			(3) JMP 3 2 (4)	JMP 3 3	LDS 3 3	LDS 4 2	(3) LDS 5 2 (6)	LDS 5 3	(4) LDX(Y) 3 3 (4)	(3) LDX(Y) 4 2 (5)	(3) LDX(Y) 5 2 (6)	(4) * 5 3 LDX(Y) (6)	E
F 1111	SEI 2 1	(5) BRCLR 7 4 (8)	BLE 3 2	SWI 14 1	CLRA 2 1	CLRB 2 1	(3) CLR 6 2 (7)	CLR 6 3	(2) XGDX(Y) 3 1 (4)	STS 4 2	(3) STS 5 2 (6)	STS 5 3	STOP 2 1	(3) STX(Y) 4 2 (5)	(3) STX(Y) 5 2 (6)	(4) * 5 3 STX(Y) (6)	F

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

***Page 3 and 4 Opcode Reference**

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to $+7.0$	V
Input Voltage	V_{in}	-0.3 to $+7.0$	V
Operating Temperature Range MC68HC11A0 MC68HC11A0V MC68HC11A0M	T_A	T_L to T_H -40 to 85 -40 to 105 -40 to 125	°C
Storage Temperature Range	T_{stg}	-55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC) Plastic 48-Pin Dual-In-Line	θ_{JA}	 50 40	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

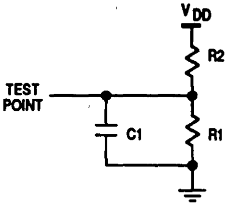
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic		Symbol	Min	Max	Unit
Output Voltage	All Outputs	V_{OL}	—	0.1	V
I_{Load} = ±10.0 µA (see Note 1)	All Outputs Except $\overline{\text{RESET}}$ and MODA	V_{OH}	$V_{DD}-0.1$	—	
Output High Voltage	All Outputs Except $\overline{\text{RESET}}$,	V_{OH}	$V_{DD}-0.8$	—	V
I_{Load} = −0.8 mA, V_{DD} = 4.5 V (see Note 1)	XTAL, and MODA				
Output Low Voltage	All Outputs Except XTAL	V_{OL}	—	0.4	V
I_{Load} = 1.6 mA					
Input High Voltage	All Inputs Except $\overline{\text{RESET}}$	V_{IH}	$0.7 \times V_{DD}$	V_{DD}	V
	$\overline{\text{RESET}}$		$0.8 \times V_{DD}$	V_{DD}	
Input Low Voltage	All Inputs	V_{IL}	V_{SS}	$0.2 \times V_{DD}$	V
I/O Ports, Three-State Leakage	PA7, PC0-PC7, PD0-PD5,	I_{OZ}	—	±10	µA
$V_{in} = V_{IH}$ or V_{IL}	AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$				
Input Current (see Note 2)		I_{in}			µA
$V_{in} = V_{DD}$ or V_{SS}	PA0-PA2, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$		—	±1	
$V_{in} = V_{DD}$ or V_{SS}	MODB/V_{STBY}		—	±10	
RAM Standby Voltage	Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	µA
Total Supply Current (see Note 3)					
RUN:		I_{DD}			
Single Chip			—	15	mA
Expanded Multiplexed			—	27	mA
WAIT:		WI_{DD}			
All Peripheral Functions Shut Down					
Single-Chip Mode			—	6	mA
Expanded Multiplexed Mode			—	10	mA
STOP:		SI_{DD}			
No Clocks, Single-Chip Mode			—	100	µA
Input Capacitance	PA0-PA2, PE0-PE7, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL	C_{in}	—	8	pF
	PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$		—	12	
Power Dissipation	Single-Chip Mode	P_D	—	85	mW
	Expanded-Multiplexed Mode		—	150	

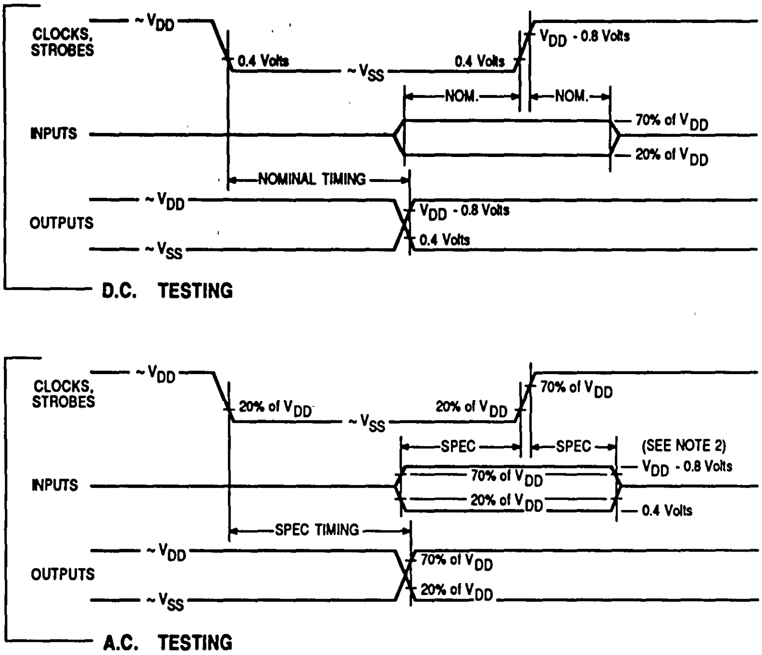
NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL} \leq 0.2$ V,
   $V_{IH} \geq V_{DD} - 0.2$ V,
   No dc loads,
   XTAL is driven with a square wave, and
   t_{cyc} = 476.5 ns.


VDD
R2
TEST POINT
C1
R1

Equivalent Test Load[1]

Pins	R1	R2	C1
PA3-PA7 PB0-PB7 PC0-PC7 PD0, PD5 E, AS, R/W	3.26K	2.38K	90pF
PD1-PD4	3.26K	2.38K	200pF


CLOCKS, STROBES
~VDD
0.4 Volts
~VSS
0.4 Volts
VDD - 0.8 Volts
NOM.
NOM.
INPUTS
70% of VDD
20% of VDD
NOMINAL TIMING
~VDD
OUTPUTS
~VSS
VDD - 0.8 Volts
0.4 Volts
D.C. TESTING
CLOCKS, STROBES
~VDD
20% of VDD
~VSS
20% of VDD
70% of VDD
SPEC
SPEC
(SEE NOTE 2)
INPUTS
70% of VDD
20% of VDD
VDD - 0.8 Volts
0.4 Volts
SPEC TIMING
~VDD
OUTPUTS
~VSS
70% of VDD
20% of VDD
A.C. TESTING

NOTES:
1. Full test loads are applied during all ac electrical test and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and $V_{DD}-0.8$ volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

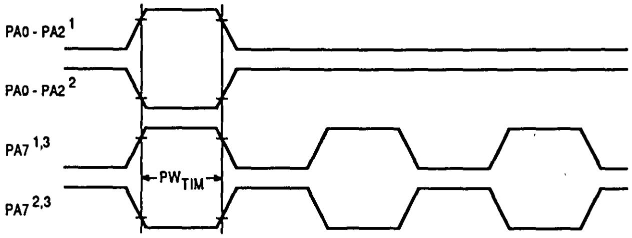
Figure 8. Test Methods

CONTROL TIMING (V_{DD}=5.0 Vdc±10%, V_{SS}=0 Vdc, $T_A=T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and12) $t_{PCS}=1/4\ t_{cyc} - 50$ ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector) (Minimum Input Time; May be Preempted by Internal Reset)	PW_{RSTL}	8 1	— —	8 1	— —	8 1	— —	t_{cyc}
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) $PW_{IRQ}=t_{cyc}+20$ ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) $PW_{TIM}=t_{cyc}+20$ ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:

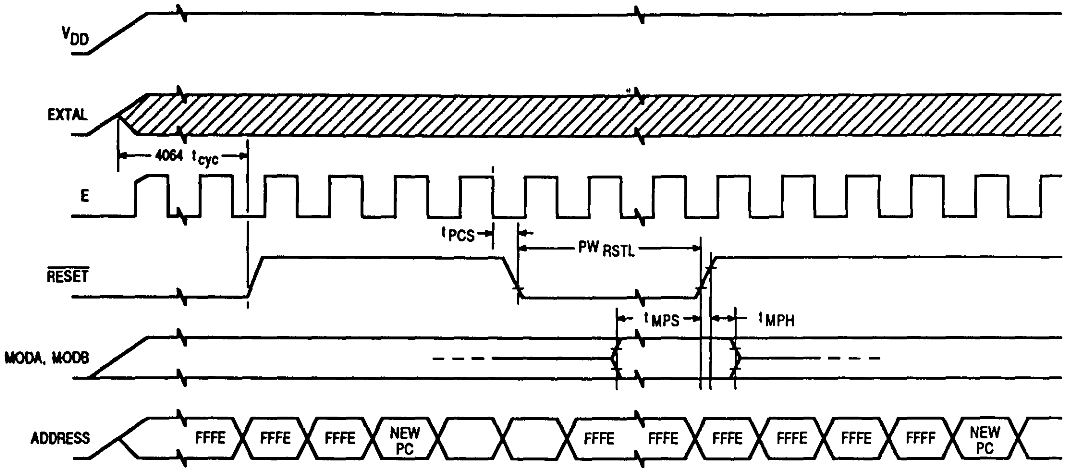
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



NOTES:

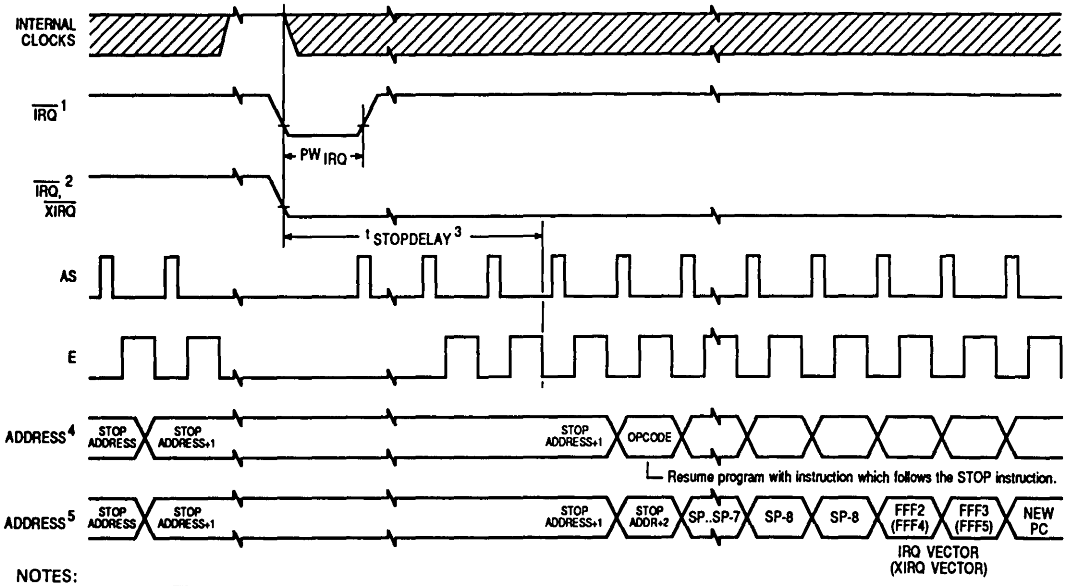
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 9. Timer Inputs Timing Diagram


VDD
EXTAL
4064 tcyc
E
tPCS
PW RSTL
RESET
tMPS
tMPH
MODA, MODB
ADDRESS
FFFE
FFFE
FFFE
NEW PC
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
NEW PC

NOTE: Refer to Table 9-7 for pin states during $\overline{\text{RESET}}$

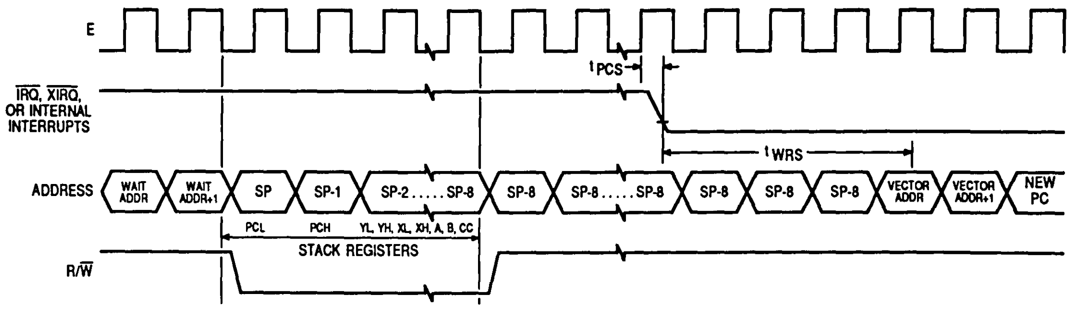
Figure 10. POR External Reset Timing Diagram


INTERNAL CLOCKS
IRQ 1
PW IRQ
IRQ, 2 XIRQ
t STOPDELAY 3
AS
E
ADDRESS 4
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
OPCODE
Resume program with instruction which follows the STOP instruction.
ADDRESS 5
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
STOP ADDR+2
SP..SP-7
SP-8
SP-8
FFF2 (FFF4)
FFF3 (FFF5)
NEW PC
IRQ VECTOR
(XIRQ VECTOR)

NOTES:

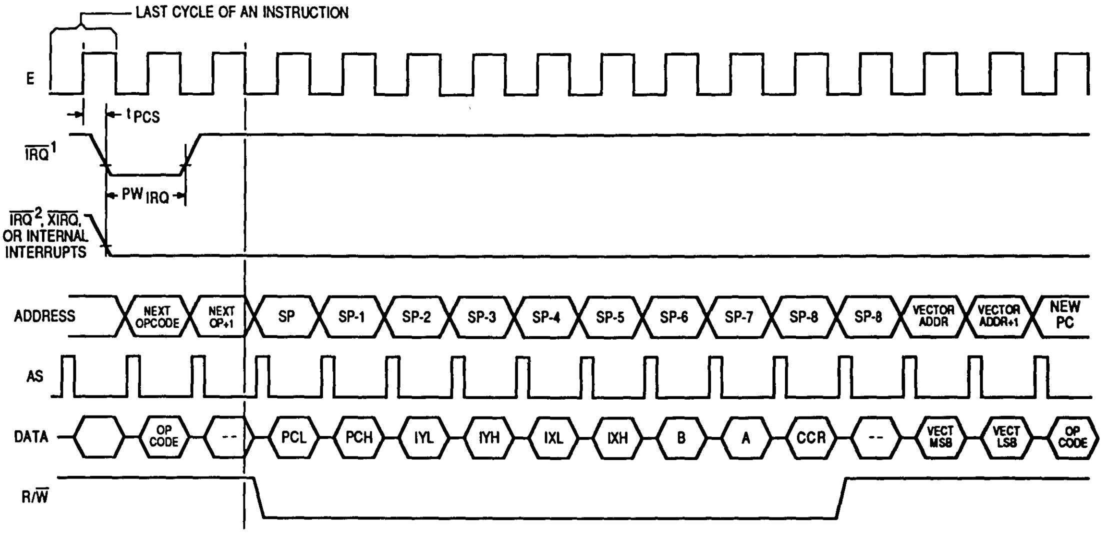
1. Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY}$ = 4064 t_{cyc} if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{\text{XIQ}}$ with X bit in CCR = 1.
5. $\overline{\text{IRQ}}$ or ($\overline{\text{XIRQ}}$ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram


E
IRQ, XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
R/W
tPCS
tWRS
WAIT ADDR
WAIT ADDR+1
SP
SP-1
SP-2.....SP-8
SP-8
SP-8.....SP-8
SP-8
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
PCL
PCH
YL, YH, XL, XH, A, B, CC
STACK REGISTERS

NOTES:
1. Refer to Table 9-7 for pin states during WAIT.
2. RESET will also cause recovery from WAIT.

Figure 12. WAIT Recovery from Interrupt Timing Diagram


LAST CYCLE OF AN INSTRUCTION
E
tPCS
IRQ1
PWIRQ
IRQ2, XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
NEXT OPCODE
NEXT OP+1
SP
SP-1
SP-2
SP-3
SP-4
SP-5
SP-6
SP-7
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
AS
DATA
OP CODE
--
PCL
PCH
IYL
IYH
IXL
IXH
B
A
CCR
--
VECT MSB
VECT LSB
OP CODE
R/W

NOTES.
1. Edge sensitive IRQ pin (IRQE bit = 1)
2. Level sensitive IRQ pin (IRQE bit = 0).

Figure 13. Interrupt Timing Diagram

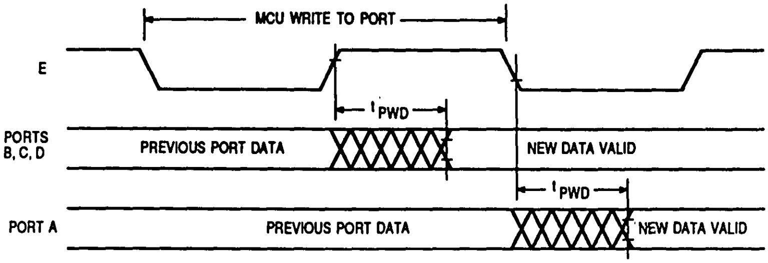
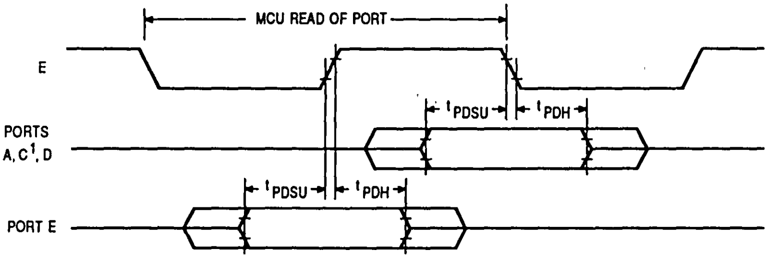

MCU WRITE TO PORT
E
t_{PWD}
PORTS B, C, D
PREVIOUS PORT DATA
NEW DATA VALID
t_{PWD}
PORT A
PREVIOUS PORT DATA
NEW DATA VALID

Figure 14. Port Write Timing Diagram


MCU READ OF PORT
E
t_{PDSU}
t_{PDH}
PORTS A, C[1], D
t_{PDSU}
t_{PDH}
PORT E

NOTE1. For non-latched operation of Port C.

Figure 15. Port Read Timing Diagram

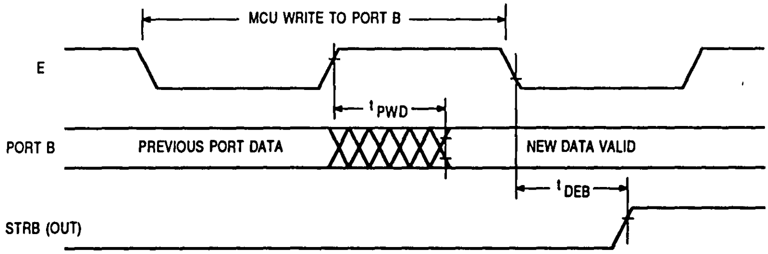

MCU WRITE TO PORT B
E
t_{PWD}
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
t_{DEB}
STRB (OUT)

Figure 16. Simple Output Strobe Timing Diagram

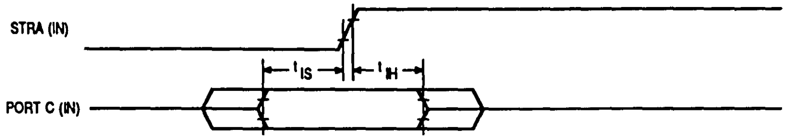
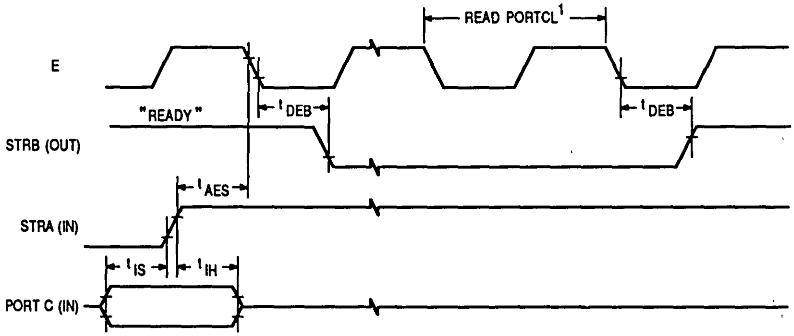
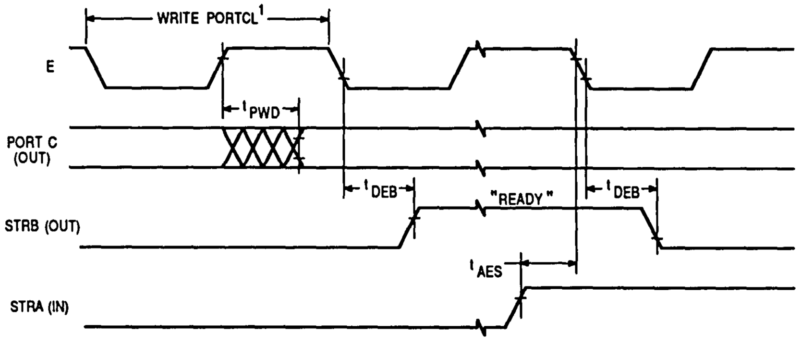

STRA (IN)
tIS
tIH
PORT C (IN)

Figure 17. Simple Input Strobe Timing Diagram


READ PORTCL1
E
"READY"
tDEB
tDEB
STRB (OUT)
tAES
STRA (IN)
tIS
tIH
PORT C (IN)

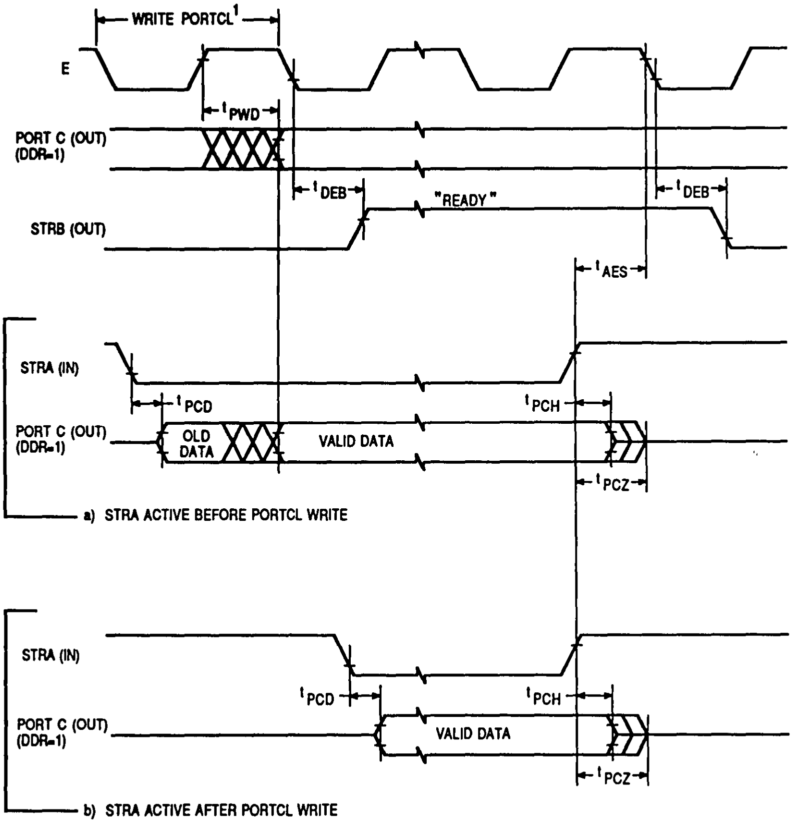
NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 18. Port C Input Handshake Timing Diagram


WRITE PORTCL1
E
tPWD
PORT C (OUT)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)

NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 19. Port C Output Handshake Timing Diagram


WRITE PORTCL[1]
E
t_{PWD}
PORT C (OUT)
(DDR=1)
t_{DEB}
"READY"
t_{DEB}
STRB (OUT)
t_{AES}
STRA (IN)
t_{PCD}
t_{PCH}
PORT C (OUT)
(DDR=1)
OLD DATA
VALID DATA
t_{PCZ}
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
t_{PCD}
t_{PCH}
PORT C (OUT)
(DDR=1)
VALID DATA
t_{PCZ}
b) STRA ACTIVE AFTER PORTCL WRITE

NOTES:

1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 15, 17, and 18)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D t_{PWD} = 1/4 t_{cyc} + 90 ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 16 and 17)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 16 and 17)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB t_{DEB} = 1/4 t_{cyc} + 100 ns (See Figure 15, 17, 18, and 19)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 17, 18, 19)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 19)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 19)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 19)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:

1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	$V_{DD} + 0.1$	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	$V_{SS} - 0.1$	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion: a. E Clock b. Internal RC Oscillator	 — —	 32 —	 — $t_{cyc} + 32$	 t_{cyc} µs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time: a. E Clock b. Internal RC Oscillator	 — —	 12 —	 — 12	 t_{cyc} µs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7 V_{RL}, V_{RH}	— —	— —	400 1.0	nA µA

NOTES:

1. Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%.

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns$)$ see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns$)$ see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

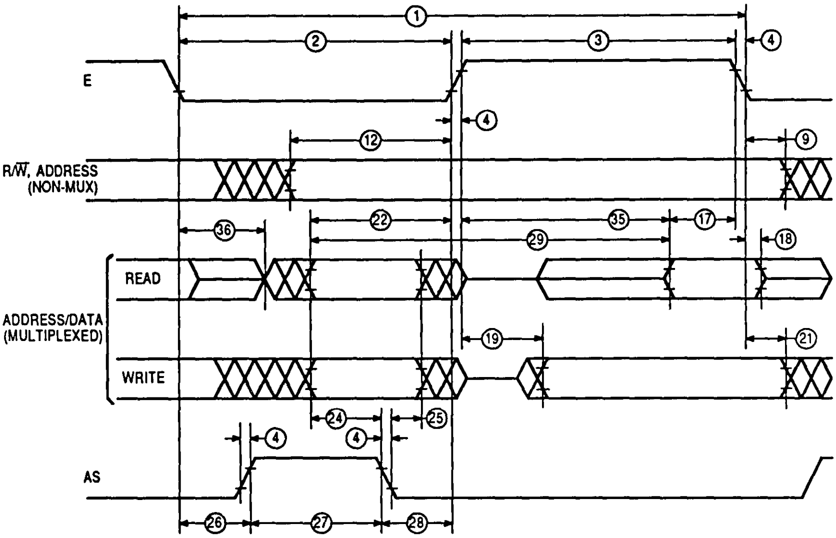
NOTES:

1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1/8 t_{cyc} in the above formulas where applicable:
   (a) (1-DC) × 1/4 t_{cyc}
   (b) DC × 1/4 t_{cyc}
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 22)

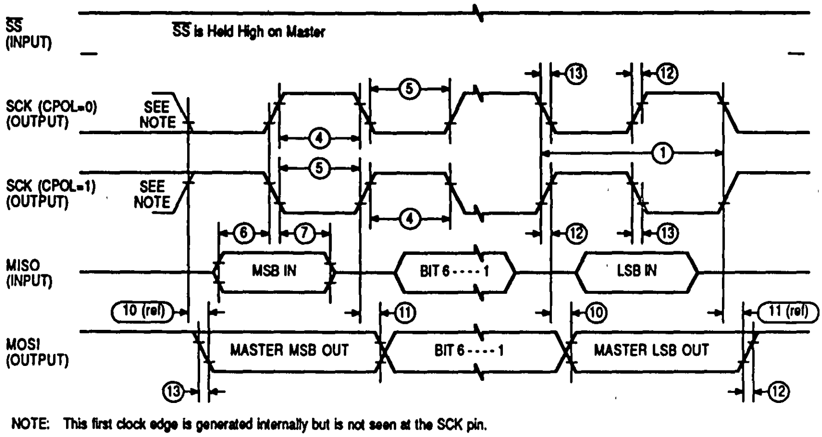
Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid (After Enable Edge)**	tv(s)	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	tho	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	trs	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	tfs	—	2.0	µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.
NOTE:
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

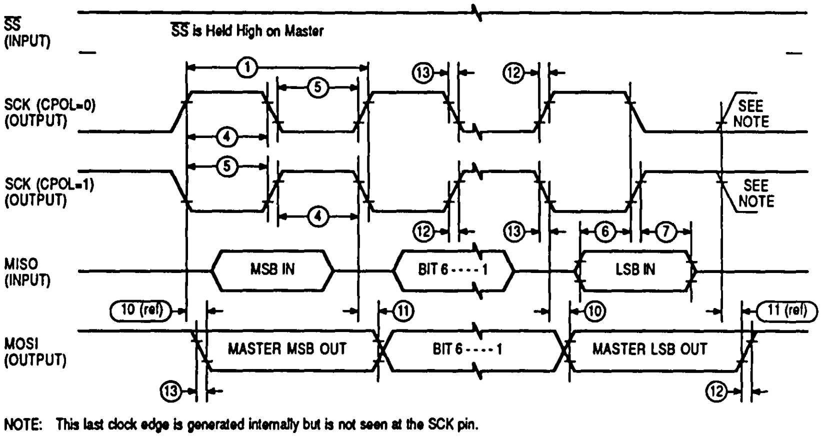

E
R/W̄, ADDRESS (NON-MUX)
ADDRESS/DATA (MULTIPLEXED)
READ
WRITE
AS
①
②
③
④
⑨
⑫
⑰
⑱
⑲
㉑
㉒
㉔
㉕
㉖
㉗
㉘
㉙
㉟
㊱

NOTE: Measurement points shown are 20% and 70% V_{DD}.

Figure 21. Expansion Bus Timing Diagram

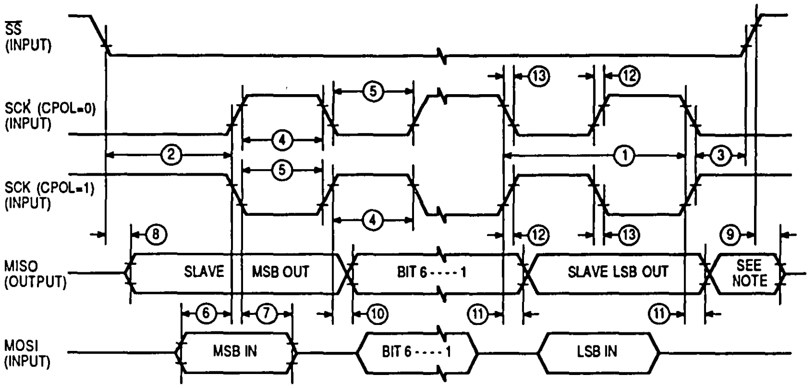

SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6----1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6----1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬
NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6----1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6----1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬
NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

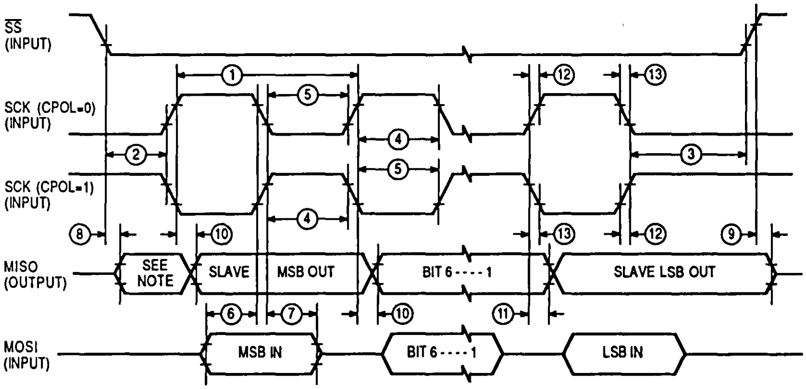
b) SPI MASTER TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6----1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6----1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE. Not defined but normally MSB of character just received

c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6----1
SLAVE LSB OUT
MSB IN
BIT 6----1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE. Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

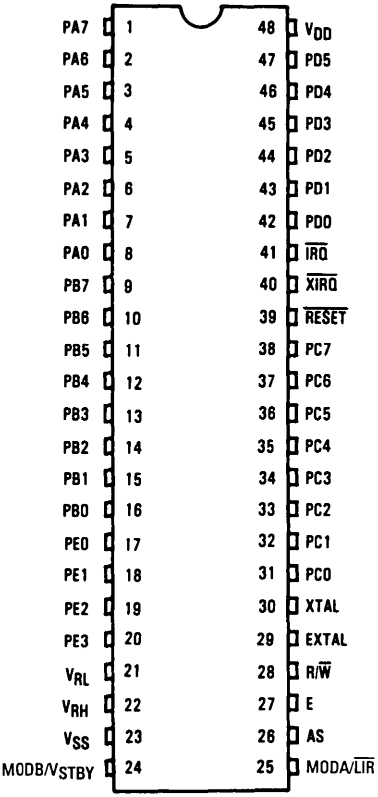
ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC11A8 HCMOS single-chip microcontroller devices.

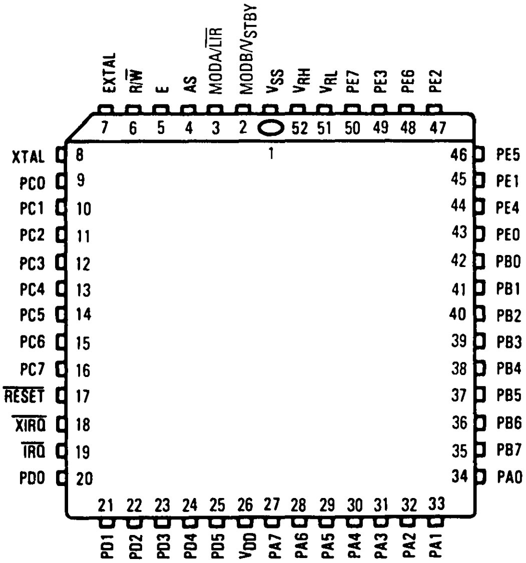
Package Type	Temperature	CONF	Description	MC Part Number
Plastic (P Suffix)	−40° to +85°C	$0C	No ROM, No EEPROM	MC68HC11A0P
PLCC (FN Suffix)	−40° to +85°C	$0C	No ROM, No EEPROM	MC68HC11A0FN

PIN ASSIGNMENTS

48-Pin Dual-in-Line Package


PA7 1
PA6 2
PA5 3
PA4 4
PA3 5
PA2 6
PA1 7
PA0 8
PB7 9
PB6 10
PB5 11
PB4 12
PB3 13
PB2 14
PB1 15
PB0 16
PE0 17
PE1 18
PE2 19
PE3 20
V_{RL} 21
V_{RH} 22
V_{SS} 23
MODB/V_{STBY} 24
48 V_{DD}
47 PD5
46 PD4
45 PD3
44 PD2
43 PD1
42 PD0
41 $\overline{IRQ}$
40 $\overline{XIRQ}$
39 $\overline{RESET}$
38 PC7
37 PC6
36 PC5
35 PC4
34 PC3
33 PC2
32 PC1
31 PC0
30 XTAL
29 EXTAL
28 R/$\overline{W}$
27 E
26 AS
25 MODA/$\overline{LIR}$

52-Lead Quad Package


EXTAL 7
R/$\overline{W}$ 6
E 5
AS 4
MODA/$\overline{LIR}$ 3
MODB/V_{STBY} 2
1
V_{SS} 52
V_{RH} 51
V_{RL} 50
PE7 49
PE3 48
PE6 47
PE2 47
XTAL 8
PC0 9
PC1 10
PC2 11
PC3 12
PC4 13
PC5 14
PC6 15
PC7 16
$\overline{RESET}$ 17
$\overline{XIRQ}$ 18
$\overline{IRQ}$ 19
PD0 20
46 PE5
45 PE1
44 PE4
43 PE0
42 PB0
41 PB1
40 PB2
39 PB3
38 PB4
37 PB5
36 PB6
35 PB7
34 PA0
21 PD1
22 PD2
23 PD3
24 PD4
25 PD5
26 V_{DD}
27 PA7
28 PA6
29 PA5
30 PA4
31 PA3
32 PA2
33 PA1

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11A1

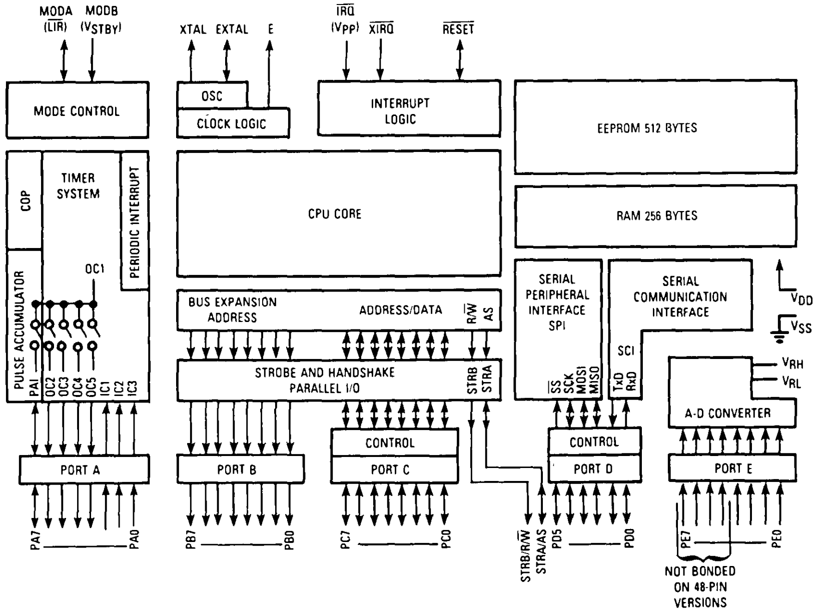
Technical Summary
8-Bit Microcontroller Unit

The MC68HC11A1 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 512 Bytes of EEPROM
- 256 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

BLOCK DIAGRAM


MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ (VPP)
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
EEPROM 512 BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
CPU CORE
RAM 256 BYTES
PULSE ACCUMULATOR
OC1
PAI OC2 OC3 OC4 OC5 IC1 IC2 IC3
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
SERIAL PERIPHERAL INTERFACE SPI
SERIAL COMMUNICATION INTERFACE
VDD
VSS
SCI
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SS SCK MOSI MISO
TxD RxD
VRH
VRL
A-D CONVERTER
CONTROL
CONTROL
PORT A
PORT B
PORT C
PORT D
PORT E
PA7 PA0
PB7 PB0
PC7 PC0
STRB/R/W
STRA/AS
PD5 PD0
PE7 PE0
NOT BONDED ON 48-PIN VERSIONS

This document contains information on a new product. Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU. This mode would not normally be used on the MC68HC11A1, because of no internal ROM.

EXPANDED MULTIPLEXED MODE

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{\text{W}}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits. This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{\text{RESET}}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins.

$\overline{\text{IRQ}}$

This pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive or level-sensitive triggering is program selectable. This pin is configured to level sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{\text{IRQ}}$.

$\overline{\text{XIRQ}}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register is set, and any interrupt is masked until enabled by software. This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{\text{LIR}}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{\text{LIR}}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The mode selections are shown below.

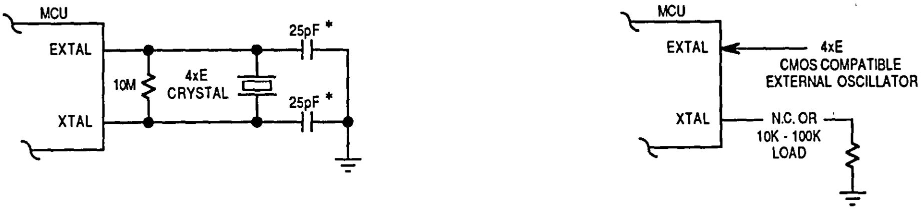
MODB	MODA	MODE SELECTED
1	0	Single Chip
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

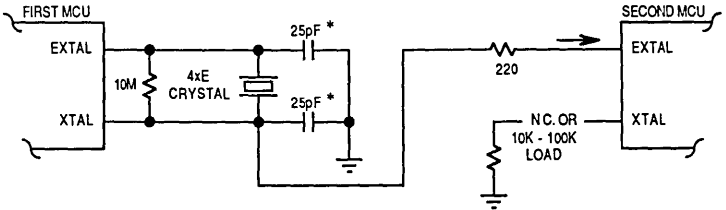
R/$\overline{\text{W}}$/STRB

This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides R/$\overline{\text{W}}$ (read-write) function. The R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.



Common Crystal Connections

External Oscillator Connections



*This value includes all stray capacitances.

One Crystal Driving Two MCUs

Figure 1. Oscillator Collections

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and $R/\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for three input capture functions; four output compare functions; and pulse accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/$\overline{SS}$	PD5$\overline{SS}$
	STRA	AS
	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4##	PE4/AN4##
E-5	PE5/AN5##	PE5/AN5##
E-6	PE6/AN6##	PE6/AN6##
E-7	PE7/AN7##	PE7/AN7##

##Not Bonded in 48-Pin Versions

parallel I/O where the STRA input and STRB output acts as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between the shaded areas (EXT) are for externally addressed memory and I/O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR A AND B

These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.

7 A 0	7 B 0
15 D 0	

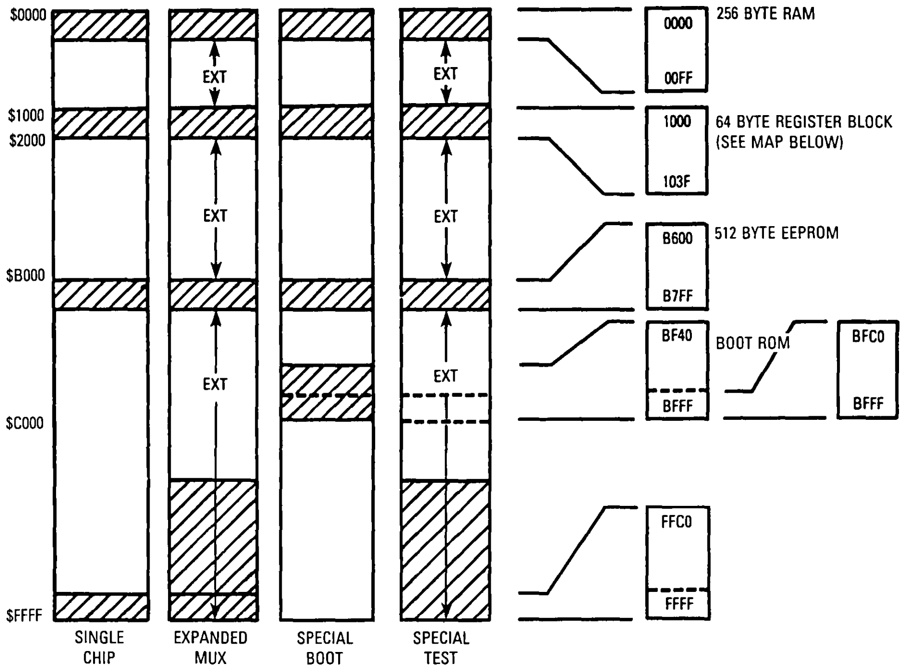
INDEX REGISTER X (IX)

This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.

15 IX 0

INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte


$0000
$1000
$2000
$B000
$C000
$FFFF
EXT
EXT
EXT
EXT
EXT
EXT
SINGLE CHIP
EXPANDED MUX
SPECIAL BOOT
SPECIAL TEST
0000
00FF
256 BYTE RAM
1000
103F
64 BYTE REGISTER BLOCK (SEE MAP BELOW)
B600
B7FF
512 BYTE EEPROM
BF40
BFFF
BOOT ROM
BFC0
BFFF
FFC0
FFFF

NOTE:

1. Either or both the internal RAM and registers can be remapped to any 4K boundary by software.
2. The EEPROM can be disabled using a control register (CONFIG), which is implemented with EEPROM cells.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TI405	Output Compare 5 Register/
$101F	Bit 7	—	—	—	—	—	—	Bit 0		Input Capture 4 Register
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	I405I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Reg 1
$1023	OC1F	OC2F	OC3F	OC4F	I405F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Reg 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Reg 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Reg 2
$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/05	RTR1	RTR0	PACTL	Pulse Accum Control Reg
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accum Count Reg
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1

Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035				PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT	EEPROM Block Protect Reg
$1036 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Circuitry
$103B	ODD	EVEN	—	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG	EEPROM Prog Control Reg.
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Reg
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F						NOCOP	ROMON	EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.

15	IY	0

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

15	PC	0

STACK POINTER (SP)

The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the accumulators A and B and registers IX and IY can be stored during certain instructions.

15	SP	0

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

7							0
S	X	H	I	N	Z	V	C

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or XIRQ) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

RESETS

The MCU can be reset four ways: 1) an active low input to the RESET pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

RESET PIN

To request an external reset, the RESET pin must be held low for eight E_{cyc} (two E_{cyc} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external RESET pin is low at the end of the power-on delay time, the processor remains in the reset condition until RESET goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the RESET pin low to reset the MCU and the external system.

The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

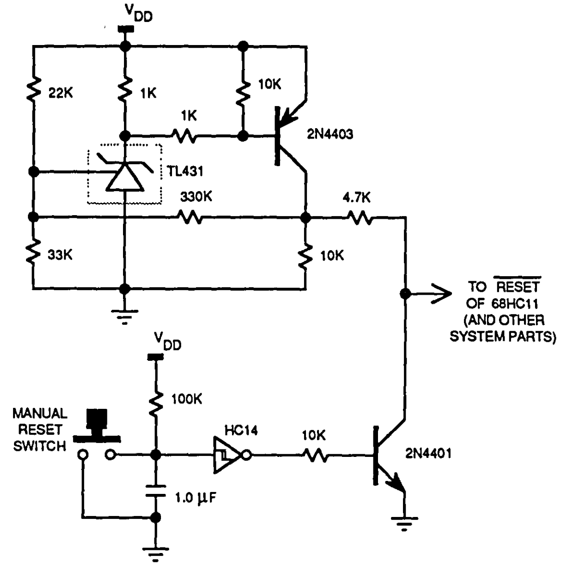
CLOCK MONITOR RESET

The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the RESET pin is driven low to reset the external system.

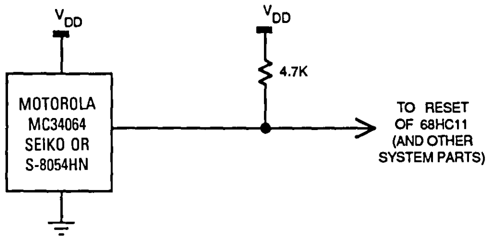
The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.

INTERRUPTS

There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the XIRQ pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is


V_{DD}
22K
1K
10K
1K
2N4403
TL431
330K
4.7K
33K
10K
TO RESET
OF 68HC11
(AND OTHER
SYSTEM PARTS)
V_{DD}
100K
MANUAL
RESET
SWITCH
HC14
10K
2N4401
1.0 µF

Reset Circuit with LVI and RC Delay


V_{DD}
V_{DD}
4.7K
MOTOROLA
MC34064
SEIKO OR
S-8054HN
TO RESET
OF 68HC11
(AND OTHER
SYSTEM PARTS)

Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

Table 2. COP Timeout Periods

CR1	CR0	$E/2^{15}$ Divided By	XTAL = 2^{23} Timeout −1/+15.6 ms	XTAL = 8.0 MHz Timeout −0/+16.4 ms	XTAL = 4.9152 MHz Timeout −0/+26.7 ms	XTAL = 4.0 MHz Timeout −0/+32.8 ms	XTAL = 3.6864 MHz Timeout −0/+35.6 ms
0	0	1	15 625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

masked during reset and upon receipt of an interrupt at the $\overline{XIRQ}$ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

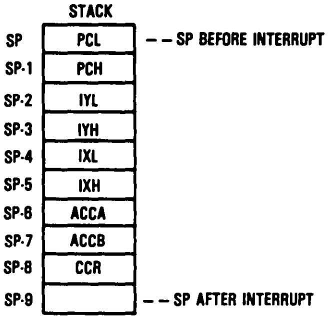


Figure 4. Stacking Order

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
*	*		
*	*		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Output Compare 5	I Bit	OC5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{IRQ}$ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	$\overline{XIRQ}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{RESET}$	None	None

I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be E/2^{13}, E/2^{14}, E/2^{15}, or E/2^{16}.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either $\overline{IRQ}$, $\overline{XIRQ}$ or $\overline{RESET}$. An external interrupt used at $\overline{IRQ}$ is only efective if the I bit in the CCR is clear. An external interrupt applied at the $\overline{XIRQ}$ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the $\overline{XIRQ}$ request. If the X bit is set, the processing will continue (if no $\overline{XIRQ}$ interrupt service routine is requested) with the instruction immediately following the STOP instruction. A low input to the $\overline{RESET}$ pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

If the internal oscillator is being used, a restart delay is required to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the $\overline{RESET}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit, and the restart delay will be imposed.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes slightly more power than the STOP mode. In the WAIT mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or $\overline{RESET}$. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the WAIT mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces WAIT-mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from \$FFFF to \$0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has three input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are three 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

TIMER CONTROL REGISTER 2 (TCTL2)

	7	6	5	4	3	2	1	0
	0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET	0	0	0	0	0	0	0	0

Bits 7-6 — Not Implemented
These bits always read zero.

EDGxB and EDGxA — Input Capture x Edge Control
These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to \$FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one.

TIMER COMPARE FORCE REGISTER (CFORC)

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

	7	6	5	4	3	2	1	0
	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET	0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action
  1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set
  0 = Has no meaning
Bits 2-0 — Not Implemented
  These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M)

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D)

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1)

	7	6	5	4	3	2	1	0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET	0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode
OL2-OL5 — Output Level
  These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TIMER INTERRUPT MASK REGISTER 1 (TMSK1)

	7	6	5	4	3	2	1	0
	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
RESET	0	0	0	0	0	0	0	0

OCxI — Output Compare x Interrupt
  1 = Interrupt sequence requested if OCxF − 1 in TFLG1
  0 = Interrupt inhibited
ICxI — Input Capture x Interrupt
  1 = Interrupt sequence requested if ICxF − 1 in TFLG1
  0 = Interrupt inhibited

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position

	7	6	5	4	3	2	1	0
	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F
RESET	0	0	0	0	0	0	0	

OCxF — Output Compare x Flag
  Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
  1 = Bit cleared
  0 = Not affected
ICxF — Input Capture x Flag
  Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
  1 = Bit cleared
  0 = Not affected

TIMER INTERRUPT MASK REGISTER 2 (TMSK2)

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG1. Two timer prescaler bits are also included in this register.

	7	6	5	4	3	2	1	0
	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled

RTII — RTI Interrupt Enable
1 = Interrupt requested when RTIF = 1
0 = RTIF interrupt disabled

PAOVI — Pulse Accumulator Overflow Interrupt Enable
1 = Interrupt requested when PAOVF = 1
0 = PAOVF disabled

PAII — Pulse Accumulator Input Interrupt Enable
1 = Interrupt requested when PAIF = 1
0 = PAIF disabled

Bits 3-2 — Not Implemented
These bits always read zero.

PR1 and PR0 — Timer Prescaler Selects
Can only be written to during initialization. Writes are disabled after the first write or after 64 E cycles out of reset.

PR1	PR0	Divide-by-Factor
0	0	1
0	1	4
1	0	8
1	1	16

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2)

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

	7	6	5	4	3	2	1	0
	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

TOF — Timer Overflow
Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag
Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag
Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag
Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented
These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0
	DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0
RESET	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
1 = Output
0 = Input only

PAEN — Pulse-Accumulator System Enable
1 = Pulse accumulator on
0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode
1 = Gated time accumulator
0 = External even counting

PEDGE — Pulse Accumulator Edge Control
This bit provides clock action along with PAMOD.
1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.
0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

Bits 3-2 — Not Implemented
These bits always read zero.

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

EEPROM PROGRAMMING

The 512 bytes of EEPROM are located at $B600 through $B7FF and have the same read cycle time as the internal ROM. Programming of the EEPROM is controlled by the EEPROM programming control register (PPROG). The EEPROM is disabled when the EEON bit in the system configuration register (CONFIG) is zero. Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz, the efficiency of this charge pump decreases, which increases the time required to program or erase a location. Recommended program and erase time is 10 milliseconds when the E clock is between 2 MHz and should be increased to as much as 20 milliseconds when E clock is between 1 MHz and 2 MHz. When E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. The following paragraphs describe how to program or erase the EEPROM using the PPROG control register.

ERASING THE EEPROM

Erasure of the EEPROM is controled by bit settings in PPROG. Programs can be written to perform bulk, row, or byte erase. In bulk erase, all 512 bytes of the EEPROM are erased. In row erase, 16 bytes ($B600-$B60F, $B610-$B61F), etc) are erased. Other MCU operations can continue to be performed during erasing provided the operations do not include reads of data from EEPROM.

PROGRAMMING EEPROM

During programming, the ROW and BYTE bits are not used. If the E clock frequency is 1 MHz or less, the CSEL bit in the OPTION register must be set. Zeros must be erased by a separate erase operation before programming. Other MCU operations can continue to be performed during programming provided the operations do not include reads of data from EEPROM.

EEPROM PROGRAMMING CONTROL REGISTER (PPROG) $103B

This 8-bit register is used to control programming and erasure of the EEPROM. This register is cleared on reset so the EEPROM is configured for normal reads.

	7	6	5	4	3	2	1	0
	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET	0	0	0	0	0	0	0	0

ODD — Program Odd Rows (TEST)
EVEN — Program Even Rows (TEST)
Bit 5 — Not Implemented
  This bit always reads zero.
BYTE — Byte Erase Select
  This bit overrides the ROW bit.
  1 = Erase only one byte
  0 = Row or bulk erase
ROW — Row Erase Select
  If BYTE bit = 1, ROW has no meaning.
  1 = Row erase
  0 = Bulk or byte erase
ERASE — Erase Mode Select
  1 = Erase mode
  0 = Normal read or program
EELAT — EEPROM Latch Control
  1 = EEPROM Address and data configured for programmming/erasing
  0 = EEPROM Address and data configured for read mode
EEPGM — EEPROM Programming Voltage Enable
  1 = Programming voltage turned on
  0 = Programming voltage turned off

NOTE

> If an attempt is made to set both the EELAT and EEPGM bit in the same write cycle, neither will be set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation currently in progress is not disturbed. If no EEPROM address is written between when EECAT is set and EEPGM is set, then no program or erase operation will take place. These safeguards were included to prevent accidental EEPROM changes in cases of program runaway. Mask set A38P, A49N, and date codes before 86xx do not have these safeguards.

ERASING THE CONFIG REGISTER

Erasing the CONFIG register follows the same procedures as that used for the EEPROM except that only bulk erase can be used on the CONFIG register. When the CONFIG register is erased, the EEPROM array is also erased. On mask set B96D, the CONFIG register may only be erased while the MCU is operating in the test or bootstrap mode.The bulk erase restriction on CONFIG is not present on all derivatives in the MC68HC11 family. Please check the applicable data sheet or technical summary for the restrictions.

PROGRAMMING THE CONFIG REGISTER

Programming the CONFIG register follows the same procedures as that used for the EEPROM except the CONFIG register address is used. On mask set B96D, the CONFIG register may only be programmed while the MCU is operating in the test or bootstrap mode.

SYSTEM CONFIGURATION REGISTER (CONFIG)

The CONFIG is implemented in EEPROM cells and controls the presence of ROM and EEPROM in the memory map and enables the COP watchdog system.

7	6	5	4	3	2	1	0
0	0	0	0		NOCOP	ROMON	EEON

Bits 7-4 — Not Implemented
  These bits are always read as zero.
Bit 3 — Not Implemented
  This bit always reads one.

NOCOP — COP System Disable
1 = COP watchdog system disabled
0 = COP watchdog system enabled

ROMON — Enable On-Chip ROM
This bit is programmed to "zero", disabling the 8K ROM. The 8K ROM memory space becomes externally accessed space. Changing this bit leads to accessing an undefined ROM.

EEON — Enable On-Chip EEPROM
When this bit is programmed to "zero", the 512-byte EEPROM is disabled, and that memory space becomes externally accessed space.

SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins: PD0 for receive data (RxD) and PD1 for the transmit data (TxD). The baud-rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/reception of a message;
2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR)

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1)

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

	7	6	5	4	3	2	1	0
	R8	T8	0	M	WAKE	0	0	0
RESET	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.

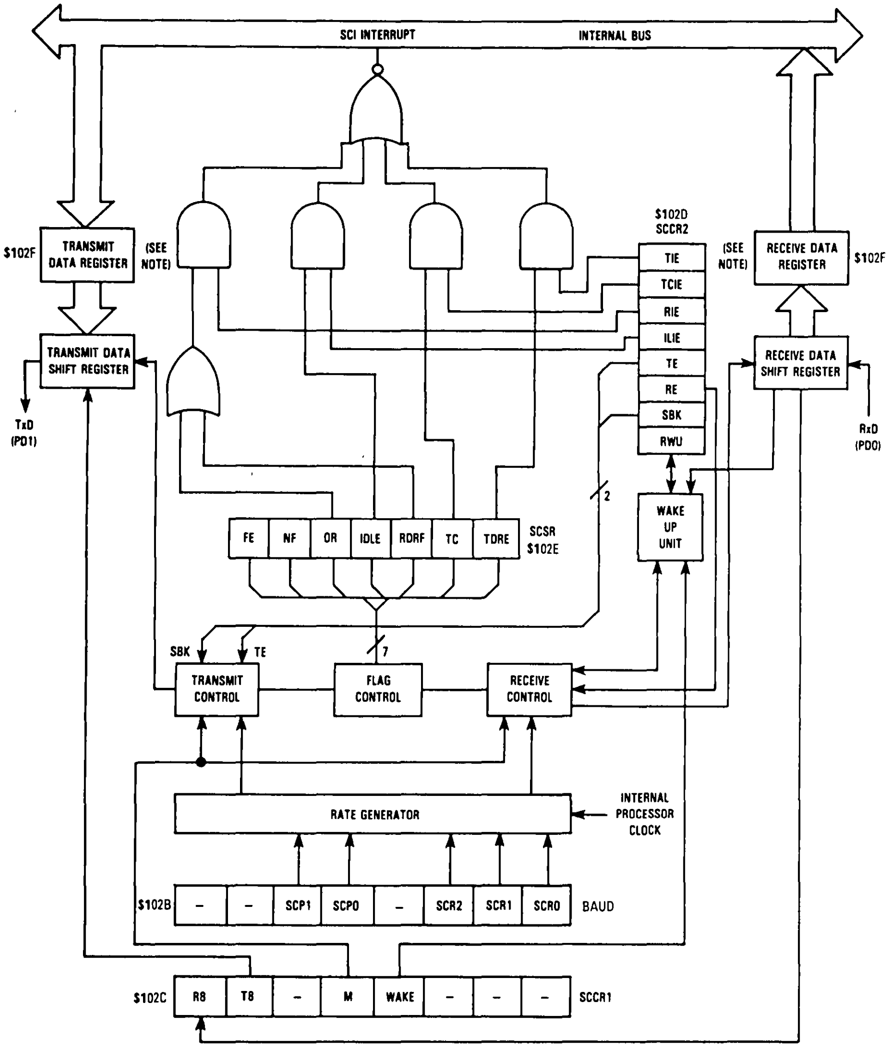
T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.

Bit 5 — Not Implemented
This bit always reads zero.

M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line

Bits 2-0 — Not Implemented
These bits always read zero.


SCI INTERRUPT
INTERNAL BUS
$102F
TRANSMIT DATA REGISTER
(SEE NOTE)
TRANSMIT DATA SHIFT REGISTER
TxD (PD1)
$102D
SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(SEE NOTE)
RECEIVE DATA REGISTER
$102F
RECEIVE DATA SHIFT REGISTER
RxD (PD0)
2
WAKE UP UNIT
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR
$102E
7
SBK
TE
TRANSMIT CONTROL
FLAG CONTROL
RECEIVE CONTROL
RATE GENERATOR
INTERNAL PROCESSOR CLOCK
$102B
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
BAUD
$102C
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE: The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal It is the transmit data register when written and received data register when read.

Figure 5. SCI Block Diagram

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt if TDRE = 1
0 = TDR interrupts disabled

TCIE — Transmit-Complete Interrupt Enable
1 = SCI interrupt if TC = 1
0 = TC interrupts disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt if RDRF or OR = 1
0 = RDRF or OR interrupt disabled

ILIE — Idle-Line Interrupt Enable
- 1 = SCI interrupt if IDLE = 1
- 0 = IDLE interrupts disabled

TE — Transmit Enable
- 1 = Transmit shift register output is applied to the TxD line
- 0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
- 1 = Receiver enabled
- 0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited

RWU — Receiver Wake Up

When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the WAKE bit is zero, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If this bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET	1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty
- 1 = Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register
- 0 = Cleared by a read of SCSR (with TDRE = 1) followed by a write to SCDR

TC — Transmit Complete
- 1 = Automatically set when all data frame, preamble, or break condition transmissions are complete
- 0 = Cleared by a read of SCSR (with TC = 1) followed by a write to SCDR

RDRF — Receive Data Register Full
- 1 = Automatically set when a character is transferred from the receiver shift register to the SCDR
- 0 = Cleared by a read of SCSR (with RDRF = 1) followed by a read of SCDR

IDLE — Idle-Line Detect

This bit is inhibited while RWU = 1.
- 1 = Automatically set when the receiver serial input becomes idle after having been active
- 0 = Cleared by a read of SCSR (with IDLE = 1) followed by a read of SCDR

OR — Overrun Error
- 1 = Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read
- 0 = Cleared by a read of SCSR (with OR = 1) followed by a read of SCDR

NF — Noise Flag
- 1 = Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame
- 0 = Cleared by a read of SCSR (with NF = 1) followed by a write to SCDR

FE — Framing Error
- 1 = Automatically set when a logic 0 is detected where a stop bit was expected
- 0 = Cleared by a read of SCSR (with FE = 1) followed by a read of SCDR

Bit 0 — Not Implemented

This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

	7	6	5	4	3	2	1	0
	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET	0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)

This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.

Bit 6 — Not Implemented

This bit always reads zero.

SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects

These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)

This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43 690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

SCR2-SCR0 — SCI Baud-Rate Selects

These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable

1 = SPI interrupt if SPIF = 1
0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable

1 = SPI system on
0 = SPI system off

DWOM — Port D Wire-OR Mode Option

This bit affects all six port D pins together.

1 = Port D outputs act as open-drain outputs
0 = Port D outputs are normal CMOS outputs

MSTR — Master Mode Select

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity

This bit selects the polarity of the SCK clock.

1 = SCK line idles high
0 = SCK line idles low

CPHA — Clock Phase

This bit selects one of two fundamentally different clock protocols. Refer to Figure 7.

SPR1 and SPR0 — SPI Clock Rate Select

These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR)

	7	6	5	4	3	2	1	0
	SPIF	WCOL	0	MODF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag

1 = Automatically set when data transfer is complete between processor and external device
0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision

If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	1	0		131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	0	1	131.072 K Baud	32.768 K Baud	76 80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

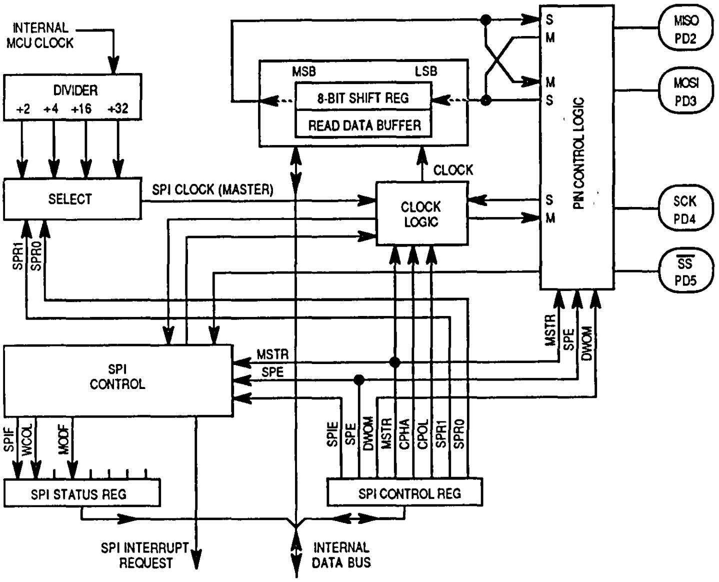

INTERNAL MCU CLOCK
DIVIDER
÷2 ÷4 ÷16 ÷32
SELECT
SPR1
SPR0
SPI CLOCK (MASTER)
MSB
LSB
8-BIT SHIFT REG
READ DATA BUFFER
CLOCK
CLOCK LOGIC
S
M
M
S
S
M
PIN CONTROL LOGIC
MISO PD2
MOSI PD3
SCK PD4
SS PD5
MSTR
SPE
DWOM
SPI CONTROL
MSTR
SPE
SPIF
WCOL
MODF
SPI STATUS REG
SPIE
SPE
DWOM
MSTR
CPHA
CPOL
SPR1
SPR0
SPI CONTROL REG
SPI INTERRUPT REQUEST
INTERNAL DATA BUS

Figure 6. SPI Block Diagram

SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{\text{SS}}$ is low and ends when the SPIF flag gets set.

1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred

0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented

This bit always reads zero.

MODF — Mode Fault

This bit indicates the possibility of a multi-master conflict for system control and therefore allows a proper exit from system operation to a reset or default system state.

1 = Automatically set when a master device has its $\overline{\text{SS}}$ pin pulled low

0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented

These bits always read zero.

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.

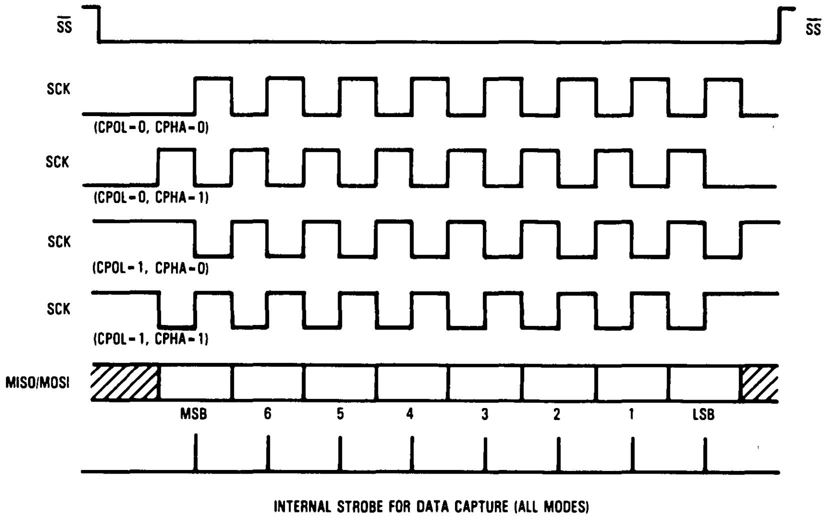


Figure 7. Data Clock Timing Diagram

2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

NOTE

In the 48-pin dual-in-line package, four conversion channels are not implemented. These include channels four through seven.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16×16	FDIV
Integer Divide 16×16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8×8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

— Continued—

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if = zero	BEQ
Branch if≥zero	BGE
Branch if>zero	BGT
Branch if Higher	BHI
Branch if≤Zero	BLE
Branch if Lower or Same	BLS
Branch if<Zero	BLT
Branch if Minus	BMI
Branch if not = Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

— Continued —

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

OPCODE MAP SUMMARY

Table 6 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

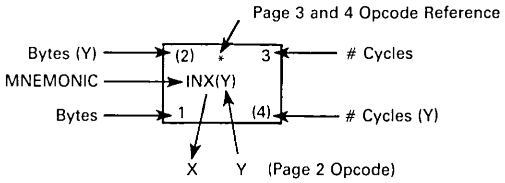
Table 6. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
HI LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI LOW
0 0000	TEST 1	2 SBA 1	3 BRA 2	(2) 3 TSX(Y) 1 (4)	2 NEGA 1	2 NEGB 1	(3) 6 NEG 2 (7)	6 NEG 3	2 SUBA 2	3 SUBA 2	(3) 4 SUBA 2 (5)	4 SUBA 3	2 SUBB 2	3 SUBB 2	(3) 4 SUBB 2 (5)	4 SUBB 3	0
1 0001	2 NOP 1	2 CBA 1	3 BRN 2	3 INS 1					2 CMPA 2	3 CMPA 2	(3) 4 CMPA 2 (5)	4 CMPA 3	2 CMPB 2	3 CMPB 2	(3) 4 CMPB 2 (5)	4 CMPB 3	1
2 0010	41 IDIV 1	6 BRSET 4	3 BHI 2	4 PULA 1					2 SBCA 2	3 SBCA 2	(3) 4 SBCA 2 (5)	4 SBCA 3	2 SBCB 2	3 SBCB 2	(3) 4 SBCB 2 (5)	4 SBCB 3	2
3 0011	41 FDIV 1	6 BRCLR 4	3 BLS 2	4 PULB 1	2 COMA 1	2 COMB 1	(3) 6 COM 2 (7)	6 COM 3	* 4 SUBD 3	* 5 SUBD 2	(3) * 6 SUBD 2 (7)	* 6 SUBD 3	4 ADDD 3	5 ADDD 2	(3) 6 ADDD 2 (7)	6 ADDD 3	3
4 0100	3 LSRD 1	6 BSET 3	3 (BHS) BCC 2	3 DES 1	2 LSRA 1	2 LSRB 1	(3) 6 LSR 2 (7)	6 LSR 3	2 ANDA 2	3 ANDA 2	(3) 4 ANDA 2 (5)	4 ANDA 3	2 ANDB 2	3 ANDB 2	(3) 4 ANDB 2 (5)	4 ANDB 3	4
5 0101	3 (LSLD) ASLD 1	6 BCLR 3	3 (BLO) BCS 2	(2) 3 TX(Y)S 1 (4)					2 BITA 2	3 BITA 2	(3) 4 BITA 2 (5)	4 BITA 3	2 BITB 2	3 BITB 2	(3) 4 BITB 2 (5)	4 BITB 3	5
6 0110	2 TAP 1	2 TAB 1	3 BNE 2	3 PSHA 1	2 RORA 1	2 RORB 1	(3) 6 ROR 2 (7)	6 ROR 3	2 LDAA 2	3 LDAA 2	(3) 4 LDAA 2 (5)	4 LDAA 3	2 LDBB 2	3 LDBB 2	(3) 4 LDBB 2 (5)	4 LDBB 3	6
7 0111	2 TPA 1	2 TBA 1	3 BEQ 2	3 PSHB 1	2 ASRA 1	2 ASRB 1	(3) 6 ASR 2 (7)	6 ASR 3		3 STAA 2	(3) 4 STAA 2 (5)	4 STAA 3		3 STBB 2	(3) 4 STBB 2 (5)	4 STBB 3	7
8 1000	(2) (3) INX(Y) (1) (4)	PAGE 2	3 BVC 2	(2) 5 PULX(Y) 1 (6)	2 ASLA 1	2 ASLB 1	(3) 6 ASL 2 (7)	6 ASL 3	2 EORA 2	(3) 3 EORA 2 (5)	4 EORA 2	4 EORA 3	2 EORB 2	3 EORB 2	(3) 4 EORB 2 (5)	4 EORB 3	8
9 1001	(2) 3 DEX(Y) 1 (4)	2 DAA 1	7 BVS 2	5 RTS 1	2 ROLA 1	2 ROLB 1	(3) 6 ROL 2 (7)	6 ROL 3	2 ADCA 2	3 ADCA 2	(3) 4 ADCA 2 (5)	4 ADCA 3	2 ADCB 2	3 ADCB 2	(3) 4 ADCB 2 (5)	4 ADCB 3	9
A 1010	2 CLV 1	PAGE 3	3 BPL 2	(2) 3 ABX(Y) 1 (4)	2 DECA 1	2 DECB 1	(3) 6 DEC 2 (7)	6 DEC 3	2 DRAA 2	3 DRAA 2	(3) 4 ORAA 2 (5)	4 ORAA 3	2 ORAB 2	3 ORAB 2	(3) 4 ORAB 2 (5)	4 ORAB 3	A
B 1011	2 SEV 1	2 ABA 1	3 BMI 2	12 RTI 1					2 ADDA 2	3 ADDA 2	(3) 4 ADDA 2 (5)	4 ADDA 3	2 ADDB 2	3 ADDB 2	(3) 4 ADDB 2 (5)	4 ADDB 3	B
C 1100	2 CLC 1	(4) 7 BSET 3 (8)	3 BGE 2	(2) 4 PSHX(Y) 1 (5)	2 INCA 1	2 INCB 1	(3) 6 INC 2 (7)	6 INC 3	4 CPX(Y) 3	5 CPX(Y) 2	(3) * 6 CPX(Y) 2 (7)	6 CPX(Y) 3	3 LDD 3	4 LDD 2	(3) 5 LDD 2 (6)	5 LDD 3	C
D 1101	2 SEC 1	(4) 7 BCLR 3 (8)	3 BLT 2	10 MUL 1	2 TSTA 1	2 TSTB 1	(3) 6 TST 2 (7)	6 TST 3	6 BSR 2	5 JSR 2	(3) 6 JSR 2 (7)	6 JSR 3	PAGE 4	4 STD 2	(3) 5 STD 2 (6)	5 STD 3	D
E 1110	2 CLI 1	(5) 7 BRSET 4 (8)	3 BGT 2	12 WAI 1			(3) 3 JMP 2 (4)	3 JMP 2	3 LDS 3	4 LDS 2	(3) 5 LDS 2 (6)	5 LDS 3	3 LDX(Y) 3	4 LDX(Y) 2	(3) 5 LDX(Y) 2 (6)	* 5 LDX(Y) 3	E
F 1111	2 SEI 1	(5) 7 BRCLR 4 (8)	3 BLE 2	14 SWI 1	2 CLRA 1	2 CLRB 1	(3) 6 CLR 2 (7)	6 CLR 3	(2) 3 XGDX(Y) 1 (4)	4 STS 2	(3) 5 STS 2 (6)	5 STS 3	2 STOP 1	(3) 4 STX(Y) 2 (5)	(3) 5 STX(Y) 2 (6)	(4) * 5 STX(Y) 3 (6)	F

*Page 3 and 4 Opcode Reference

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC11A1 MC68HC11A1V MC68HC11A1M	T_A	T_L to T_H −40 to 85 −40 to 105 −40 to 125	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC) Plastic 48-Pin Dual-In-Line	θ_{JA}	 50 40	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot {P_D}^2 \qquad (3)$$

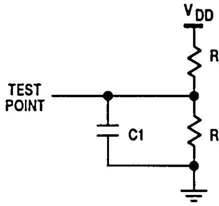
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage — All Outputs I_{Load} = ± 10.0 µA (see Note 1) — All Outputs Except $\overline{\text{RESET}}$ and MODA	V_{OL} V_{OH}	— V_{DD} − 0.1	0.1 —	V
Output High Voltage — All Outputs Except $\overline{\text{RESET}}$, XTAL, and MODA I_{Load} = − 0.8 mA, V_{DD} = 4.5 V (see Note 1)	V_{OH}	V_{DD} − 0.8	—	V
Output Low Voltage — All Outputs Except XTAL I_{Load} = 1.6 mA	V_{OL}	—	0.4	V
Input High Voltage — All Inputs Except $\overline{\text{RESET}}$ $\overline{\text{RESET}}$	V_{IH}	0.7 × V_{DD} 0.8 × V_{DD}	V_{DD} V_{DD}	V
Input Low Voltage — All Inputs	V_{IL}	V_{SS}	0.2 × V_{DD}	V
I/O Ports, Three-State Leakage — PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$ $V_{in} = V_{IH}$ or V_{IL}	I_{OZ}	—	± 10	µA
Input Current (see Note 2) $V_{in} = V_{DD}$ or V_{SS} — PA0-PA2, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ $V_{in} = V_{DD}$ or V_{SS} — MODB/V_{STBY}	I_{in}	 — —	 ± 1 ± 10	µA
RAM Standby Voltage — Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current — Powerdown	I_{SB}	—	20	µA
Total Supply Current (see Note 3)				
RUN:	I_{DD}			
Single Chip		—	15	mA
Expanded Multiplexed		—	27	mA
WAIT:	WI_{DD}			
All Peripheral Functions Shut Down				
Single-Chip Mode		—	6	mA
Expanded Multiplexed Mode		—	10	mA
STOP:	SI_{DD}			
No Clocks, Single-Chip Mode		—	100	µA
Input Capacitance — PA0-PA2, PE0-PE7, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	C_{in}	— —	8 12	pF
Power Dissipation — Single-Chip Mode Expanded-Multiplexed Mode	P_D	— —	85 150	mW

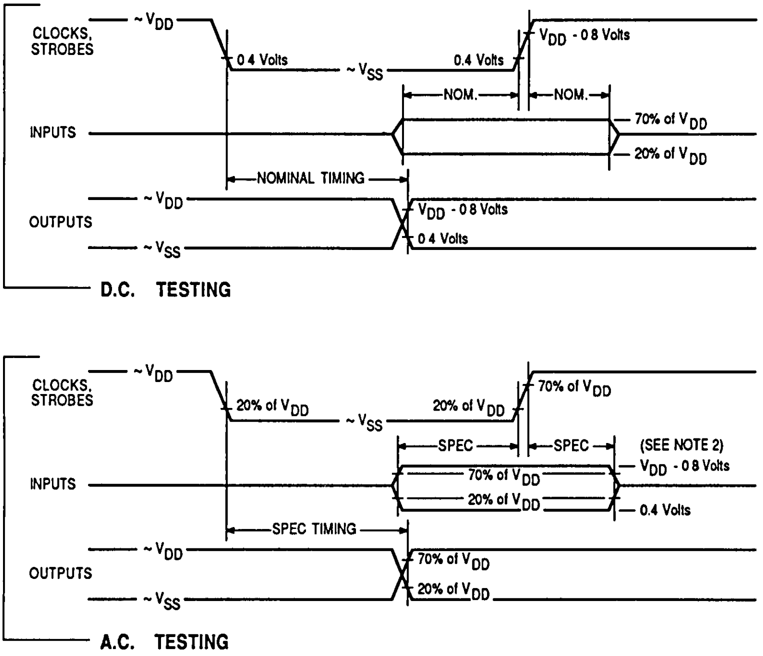
NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL} \leq 0.2$ V,
   $V_{IH} \geq V_{DD} - 0.2$ V,
   No dc loads,
   EXTAL is driven with a square wave, and
   t_{cyc} = 476.5 ns.


VDD
R2
TEST POINT
C1
R1

Equivalent Test Load[1]

Pins	R1	R2	C1
PA3-PA7 PB0-PB7 PC0-PC7 PD0, PD5 E, AS, R/W	3.26K	2.38K	90pF
PD1-PD4	3.26K	2.38K	200pF


CLOCKS, STROBES
~VDD
0.4 Volts
~VSS
0.4 Volts
VDD - 0.8 Volts
NOM.
NOM.
INPUTS
70% of VDD
20% of VDD
NOMINAL TIMING
OUTPUTS
~VDD
~VSS
VDD - 0.8 Volts
0.4 Volts
D.C. TESTING
CLOCKS, STROBES
~VDD
20% of VDD
~VSS
20% of VDD
70% of VDD
SPEC
SPEC
(SEE NOTE 2)
INPUTS
70% of VDD
20% of VDD
VDD - 0.8 Volts
0.4 Volts
SPEC TIMING
OUTPUTS
~VDD
~VSS
70% of VDD
20% of VDD
A.C. TESTING

NOTES:

1. Full test loads are applied during all ac electrical test and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and $V_{DD}-0.8$ volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

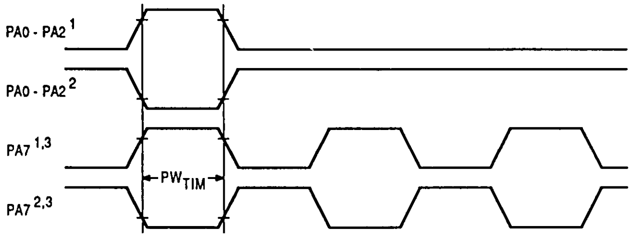
Figure 8. Test Methods

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	$4 f_o$	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and12) $t_{PCS} = 1/4\ t_{cyc} - 50$ ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time; May be Preempted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) $PW_{IRQ} = t_{cyc} + 20$ ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) $PW_{TIM} = t_{cyc} + 20$ ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:

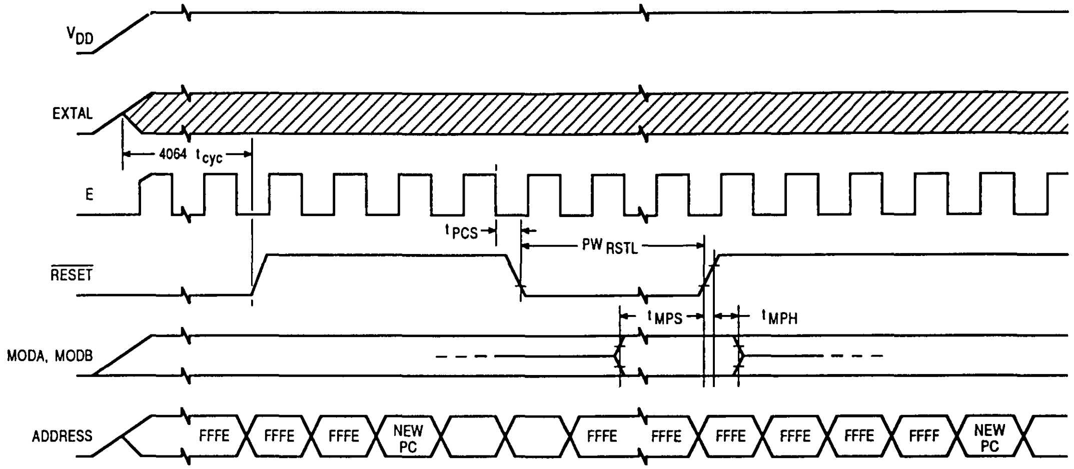
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



NOTES:

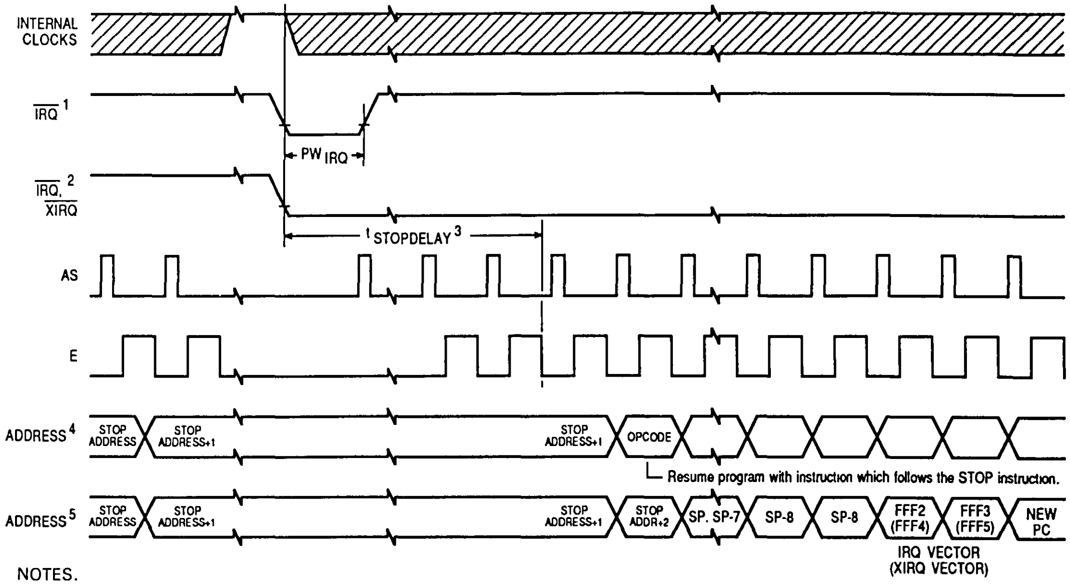
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 9. Timer Inputs Timing Diagram


VDD
EXTAL
4064 tcyc
E
tPCS
PW RSTL
RESET
tMPS
tMPH
MODA, MODB
ADDRESS
FFFE
FFFE
FFFE
NEW PC
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
NEW PC

NOTE Refer to Table 9-7 for pin states during $\overline{RESET}$

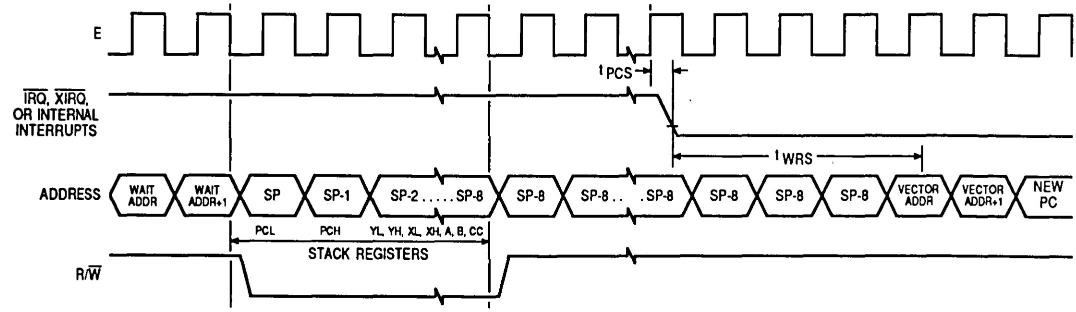
Figure 10. POR External Reset Timing Diagram


INTERNAL CLOCKS
IRQ 1
PW IRQ
IRQ, XIRQ 2
tSTOPDELAY 3
AS
E
ADDRESS 4
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
OPCODE
Resume program with instruction which follows the STOP instruction.
ADDRESS 5
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
STOP ADDR+2
SP, SP-7
SP-8
SP-8
FFF2 (FFF4)
FFF3 (FFF5)
NEW PC
IRQ VECTOR
(XIRQ VECTOR)

NOTES.

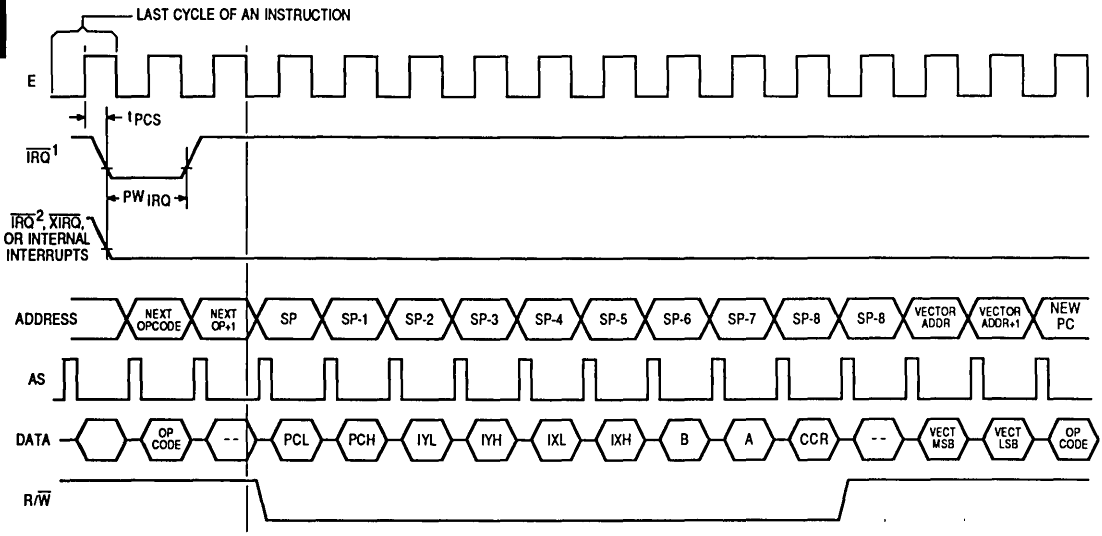
1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY} = 4064\ t_{cyc}$ if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{XIQ}$ with X bit in CCR = 1.
5. $\overline{IRQ}$ or ($\overline{XIRQ}$ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram


E
t_{PCS}
IRQ, XIRQ, OR INTERNAL INTERRUPTS
t_{WRS}
ADDRESS
WAIT ADDR
WAIT ADDR+1
SP
SP-1
SP-2.....SP-8
SP-8
SP-8.. .SP-8
SP-8
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
PCL
PCH
YL, YH, XL, XH, A, B, CC
STACK REGISTERS
R/W

NOTES:
1. Refer to Table 9-7 for pin states during WAIT.
2. RESET will also cause recovery from WAIT.

Figure 12. WAIT Recovery from Interrupt Timing Diagram


LAST CYCLE OF AN INSTRUCTION
E
t_{PCS}
IRQ¹
PW_{IRQ}
IRQ², XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
NEXT OPCODE
NEXT OP+1
SP
SP-1
SP-2
SP-3
SP-4
SP-5
SP-6
SP-7
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
AS
DATA
OP CODE
--
PCL
PCH
IYL
IYH
IXL
IXH
B
A
CCR
--
VECT MSB
VECT LSB
OP CODE
R/W

NOTES:
1. Edge sensitive IRQ pin (IRQE bit = 1)
2. Level sensitive IRQ pin (IRQE bit = 0)

Figure 13. Interrupt Timing Diagram

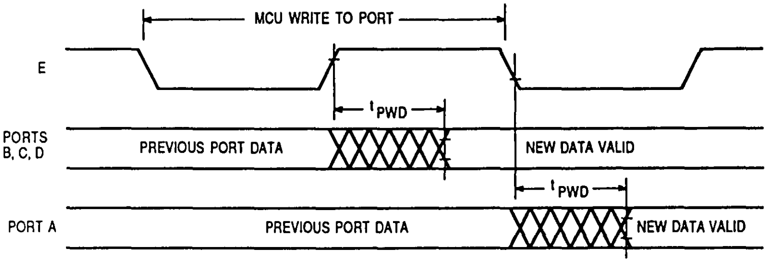
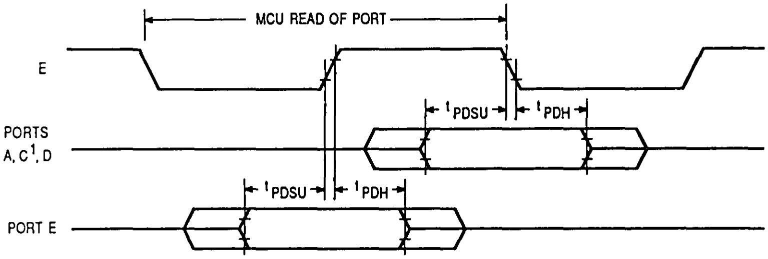

MCU WRITE TO PORT
E
tPWD
PORTS B, C, D
PREVIOUS PORT DATA
NEW DATA VALID
tPWD
PORT A
PREVIOUS PORT DATA
NEW DATA VALID

Figure 14. Port Write Timing Diagram


MCU READ OF PORT
E
tPDSU
tPDH
PORTS A, C1, D
tPDSU
tPDH
PORT E

NOTE1 For non-latched operation of Port C

Figure 15. Port Read Timing Diagram

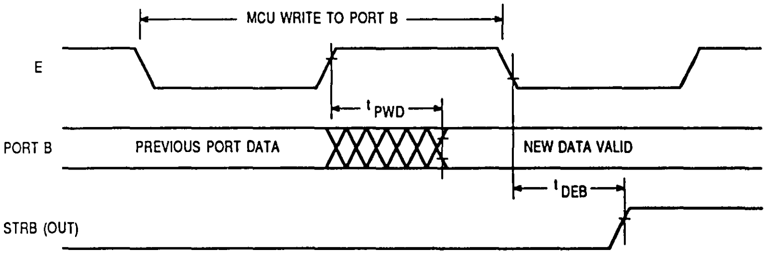

MCU WRITE TO PORT B
E
tPWD
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
tDEB
STRB (OUT)

Figure 16. Simple Output Strobe Timing Diagram

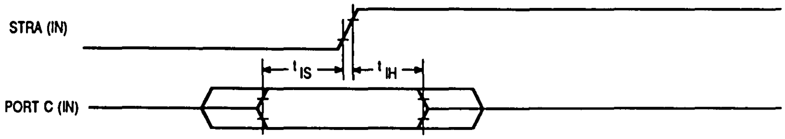
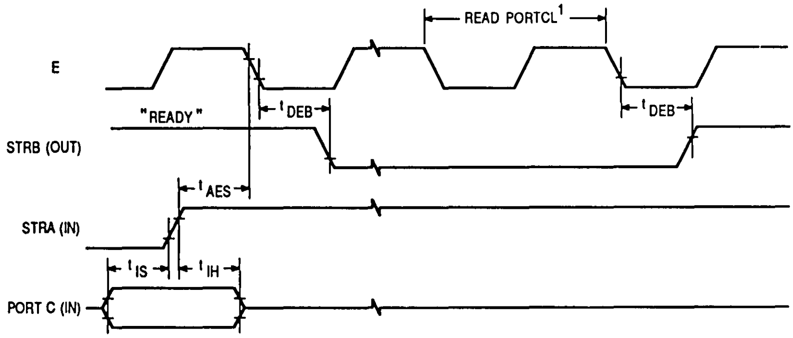
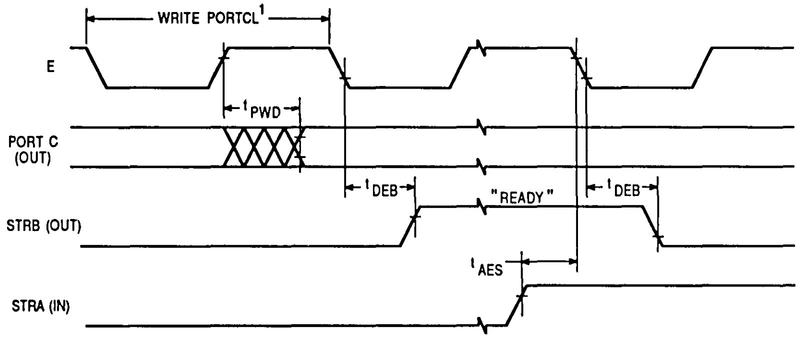

STRA (IN)
PORT C (IN)
t_{IS}
t_{IH}

Figure 17. Simple Input Strobe Timing Diagram


READ PORTCL¹
E
"READY"
t_{DEB}
t_{DEB}
STRB (OUT)
t_{AES}
STRA (IN)
t_{IS}
t_{IH}
PORT C (IN)

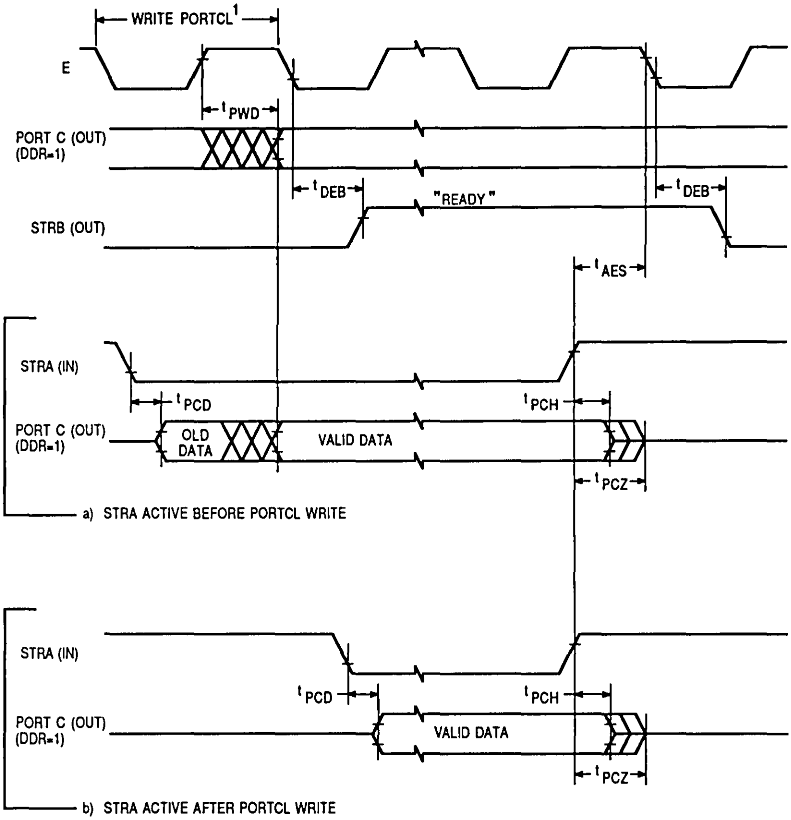
NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 18. Port C Input Handshake Timing Diagram


WRITE PORTCL¹
E
t_{PWD}
PORT C
(OUT)
t_{DEB}
"READY"
t_{DEB}
STRB (OUT)
t_{AES}
STRA (IN)

NOTES.
1 After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 19. Port C Output Handshake Timing Diagram


WRITE PORTCL1
E
tPWD
PORT C (OUT) (DDR=1)
tDEB
"READY"
STRB (OUT)
tDEB
tAES
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
OLD DATA
VALID DATA
tPCZ
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
VALID DATA
tPCZ
b) STRA ACTIVE AFTER PORTCL WRITE

NOTES.
1 After reading PIOC with STAF set
2 Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 15, 17, and 18)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D t_{PWD} = 1/4 t_{cyc} + 90 ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 16 and 17)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 16 and 17)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB t_{DEB} = 1/4 t_{cyc} + 100 ns (See Figure 15, 17, 18, and 19)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 17, 18, 19)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 19)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 19)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 19)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:

1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively)
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	$V_{DD} + 0.1$	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	$V_{SS} - 0.1$	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion: a. E Clock b. Internal RC Oscillator	 — —	 32 —	 — $t_{cyc} + 32$	 t_{cyc} μs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time: a. E Clock b. Internal RC Oscillator	 — —	 12 —	 — 12	 t_{cyc} μs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7	—	—	400	nA
	V_{RL}, V_{RH}	—	—	1.0	μA

NOTES:

1. Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%.

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns) see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns) see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES:

1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of $1/8\ t_{cyc}$ in the above formulas where applicable:
   (a) $(1\text{-DC}) \times 1/4\ t_{cyc}$
   (b) $DC \times 1/4\ t_{cyc}$
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD}=5.0 Vdc±10%, V_{SS}=0 Vdc, $T_A=T_L$ to T_H, see Figure 22)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid (After Enable Edge)**	tv(s)	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	tho	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L=200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	trs	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L=200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	tfs	—	2.0	µs

*Signal production depends on software

**Assumes 200 pF load on all SPI pins.

NOTE:

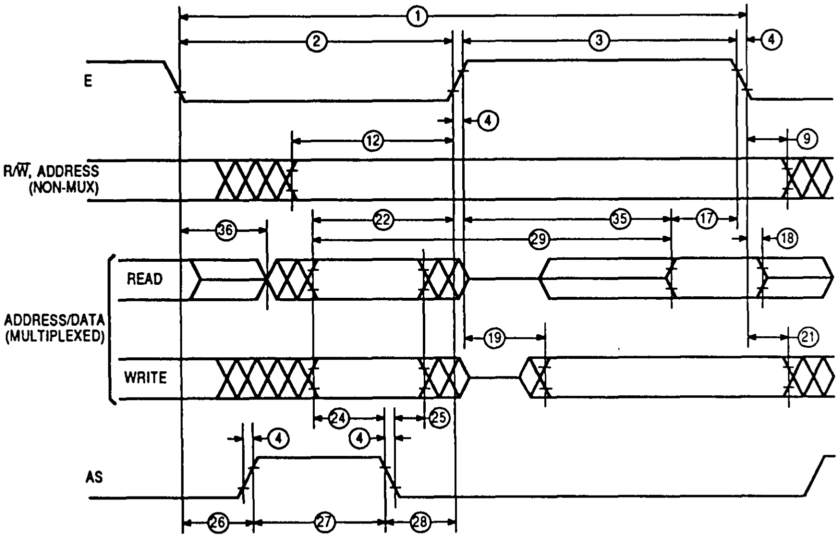
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

EEPROM CHARACTERISTICS (V_{DD}=5.0 Vdc ± 10%, V_{SS}=0 Vdc, T_A=T_L to T_H)

Characteristic		Temperature Range			Unit
		−40 to 85°C	−40 to 105°C	−40 to 125°C	
Programming Time (see Note 1)	Under 1.0 MHz with RC Oscillator Enabled	10	15	20	ms
	1.0 to 2.0 MHz with RC Oscillator Disabled	20	Must Use RC	Must Use RC	
	2.0 MHz (or Anytime RC Oscillator Enabled)	10	15	20	
Erase Time (see Note 1)	Byte, Row, and Bulk	10	10	10	ms
Write/Erase Endurance (see Note 2)		10,000	10,000	10,000	Cycles
Data Retention (see Note 2)		10	10	10	Years

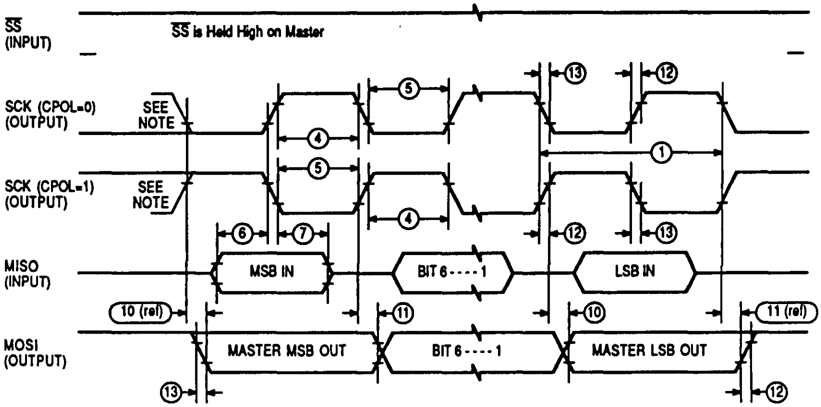
NOTES:

1. The RC oscillator must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. See current quarterly Reliability Monitor report for current failure rate information.



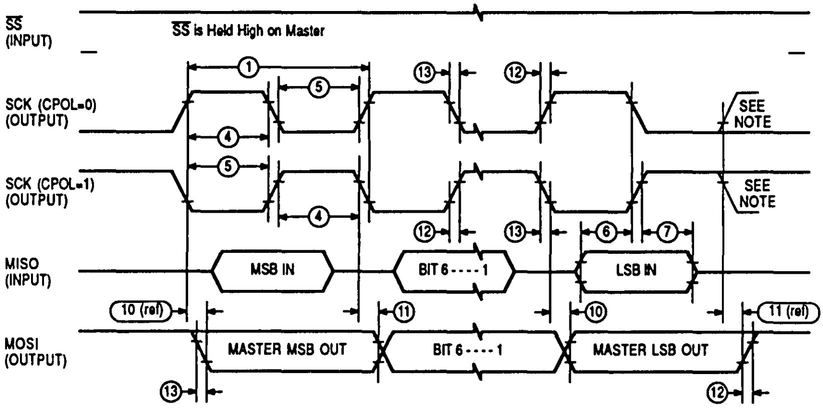
NOTE: Measurement points shown are 20% and 70% V_{DD}

Figure 21. Expansion Bus Timing Diagram


SS
(INPUT)
SS is Held High on Master
SCK (CPOL=0)
(OUTPUT)
SEE
NOTE
SCK (CPOL=1)
(OUTPUT)
SEE
NOTE
MISO
(INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI
(OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

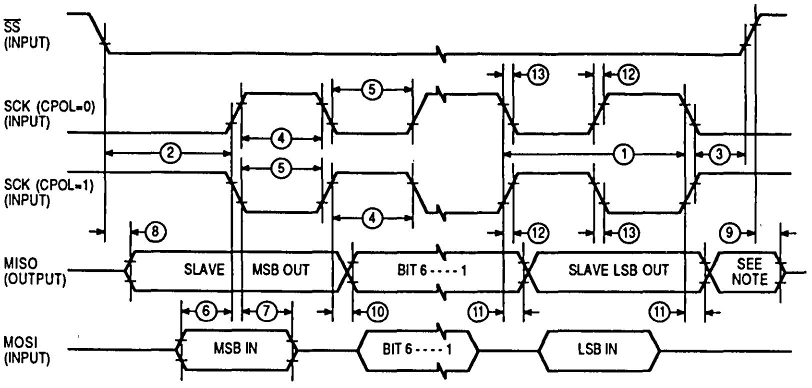
a) SPI MASTER TIMING (CPHA=0)


SS
(INPUT)
SS is Held High on Master
SCK (CPOL=0)
(OUTPUT)
SEE
NOTE
SCK (CPOL=1)
(OUTPUT)
SEE
NOTE
MISO
(INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI
(OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

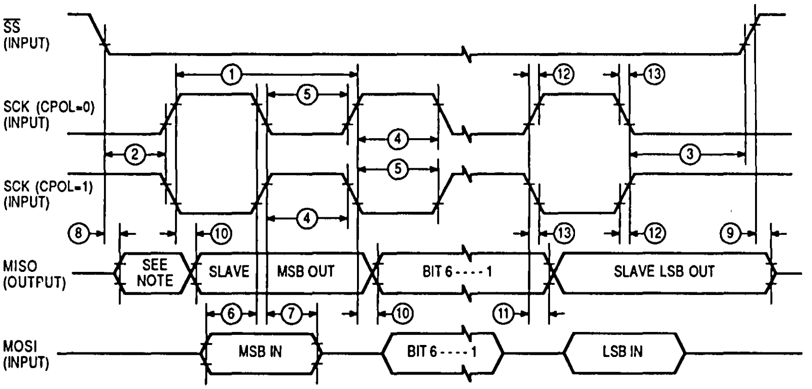
b) SPI MASTER TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE: Not defined but normally MSB of character just received

c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN

NOTE: Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

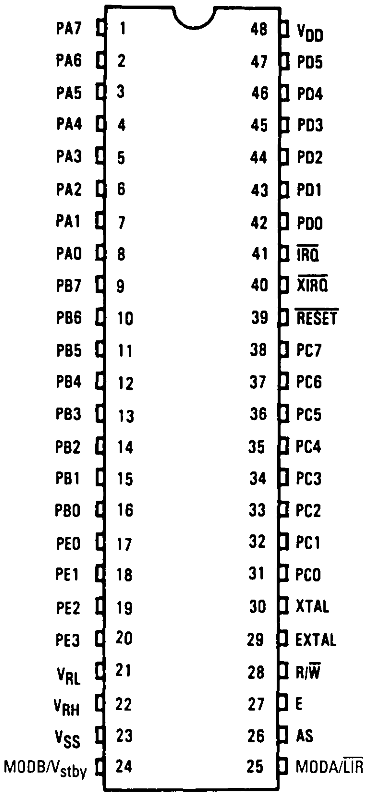
ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC11A1 HCMOS single-chip microcontroller device.

Package Type	Temperature	CONF	Description	MC Part Number
Plastic (P Suffix)	−40 to +85°C	$0D	No ROM	MC68HC11A1P
	−40 to +105°C	$0D	No ROM	MC68HC11A1VP
	−40 to +125°C	$0D	No ROM	MC68HC11A1MP
	−40 to +85°C	$09	No ROM, COP On	MC68HCP11A1P
	−40 to +105°C	$09	No ROM, COP On	MC68HCP11A1VP
	−40 to +125°C	$09	No ROM, COP On	MC68HCP11A1MP
PLCC (FN Suffix)	−40 to +85°C	$0D	No ROM	MC68HC11A1FN
	−40 to +105°C	$0D	No ROM	MC68HC11A1VFN
	−40 to 125°C	$0D	No ROM	MC68HC11A1MFN
	−40 to +85°C	$09	No ROM, COP On	MC68HCP11A1FN

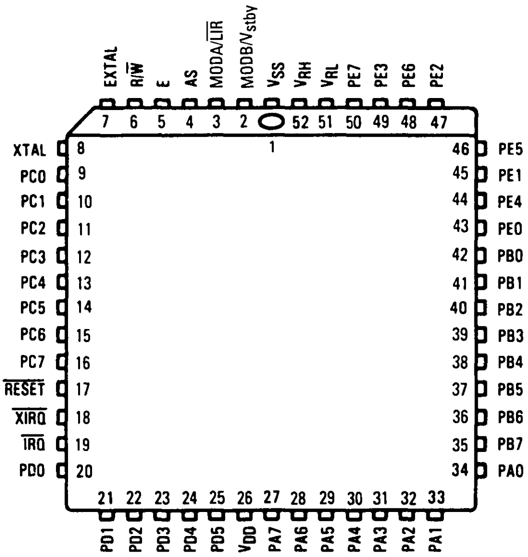
PIN ASSIGNMENTS

48-Pin Dual-in-Line Package



PA7 1 48 VDD
PA6 2 47 PD5
PA5 3 46 PD4
PA4 4 45 PD3
PA3 5 44 PD2
PA2 6 43 PD1
PA1 7 42 PD0
PA0 8 41 IRQ
PB7 9 40 XIRQ
PB6 10 39 RESET
PB5 11 38 PC7
PB4 12 37 PC6
PB3 13 36 PC5
PB2 14 35 PC4
PB1 15 34 PC3
PB0 16 33 PC2
PE0 17 32 PC1
PE1 18 31 PC0
PE2 19 30 XTAL
PE3 20 29 EXTAL
VRL 21 28 R/W
VRH 22 27 E
VSS 23 26 AS
MODB/Vstby 24 25 MODA/LIR

52-Lead Quad Package



EXTAL 7
R/W 6
E 5
AS 4
MODA/LIR 3
MODB/Vstby 2
VSS 1
VRH 52
VRL 51
PE7 50
PE3 49
PE6 48
PE2 47
XTAL 8
PC0 9
PC1 10
PC2 11
PC3 12
PC4 13
PC5 14
PC6 15
PC7 16
RESET 17
XIRQ 18
IRQ 19
PD0 20
PE5 46
PE1 45
PE4 44
PE0 43
PB0 42
PB1 41
PB2 40
PB3 39
PB4 38
PB5 37
PB6 36
PB7 35
PA0 34
PD1 21
PD2 22
PD3 23
PD4 24
PD5 25
VDD 26
PA7 27
PA6 28
PA5 29
PA4 30
PA3 31
PA2 32
PA1 33

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11A8

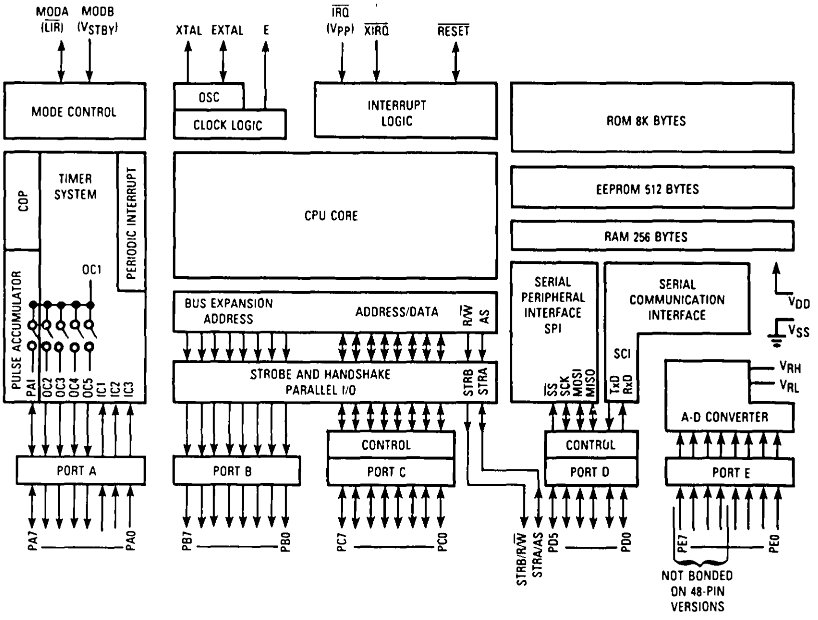
Technical Summary
8-Bit Microcontroller

The MC68HC11A8 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 8K Bytes of ROM
- 512 Bytes of EEPROM
- 256 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

BLOCK DIAGRAM


MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ (VPP)
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
ROM 8K BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
CPU CORE
EEPROM 512 BYTES
RAM 256 BYTES
PULSE ACCUMULATOR
OC1
PAI
OC2
OC3
OC4
OC5
IC1
IC2
IC3
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
SERIAL PERIPHERAL INTERFACE SPI
SERIAL COMMUNICATION INTERFACE
VDD
VSS
SCI
VRH
VRL
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SS
SCK
MOSI
MISO
TxD
RxD
A-D CONVERTER
CONTROL
CONTROL
PORT A
PORT B
PORT C
PORT D
PORT E
PA7
PA0
PB7
PB0
PC7
PC0
STRB/R/W
STRA/AS
PD5
PD0
PE7
PE0
NOT BONDED ON 48-PIN VERSIONS

This document contains information on a new product. Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE (MODE0)

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU.

EXPANDED MULTIPLEXED MODE (MODE1)

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The $R/\overline{W}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits. This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{\text{RESET}}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins.

$\overline{\text{IRQ}}$

This pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive or level-sensitive triggering is program selectable. This pin is configured to level-sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{\text{IRQ}}$.

$\overline{\text{XIRQ}}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register is set, and any interrupt is masked until enabled by software. This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{\text{LIR}}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{\text{LIR}}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The mode selections are shown below.

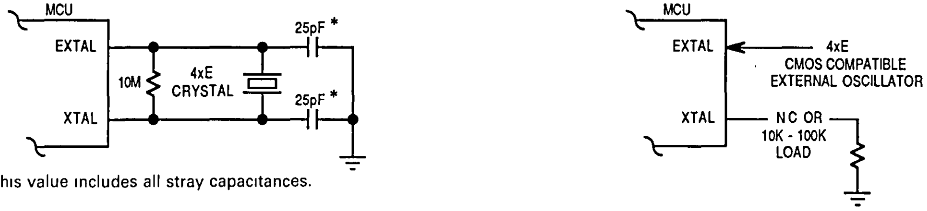
MODB	MODA	MODE SELECTED
1	0	Single Chip
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

$R/\overline{W}$/STRB

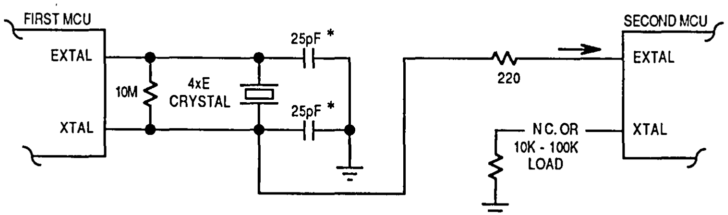
This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides $R/\overline{W}$ (read-write) function. The $R/\overline{W}$ is used to control the direction of transfers on the external data bus.



*This value includes all stray capacitances.

Common Crystal Connections

External Oscillator Connections



*This value includes all stray capacitances

One Crystal Driving Two MCUs

Figure 1. Oscillator Collections

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for three input capture functions; four output compare functions; and pulse accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/$\overline{SS}$	PD5$\overline{SS}$
	STRA	AS
	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4##	PE4/AN4##
E-5	PE5/AN5##	PE5/AN5##
E-6	PE6/AN6##	PE6/AN6##
E-7	PE7/AN7##	PE7/AN7##

##Not Bonded in 48-Pin Versions

parallel I/O where the STRA input and STRB output acts as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between the shaded areas (EXT) are for externally addressed memory and I/O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR A AND B

These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.

7	A	0	7	B	0
15		D			0

INDEX REGISTER X (IX)

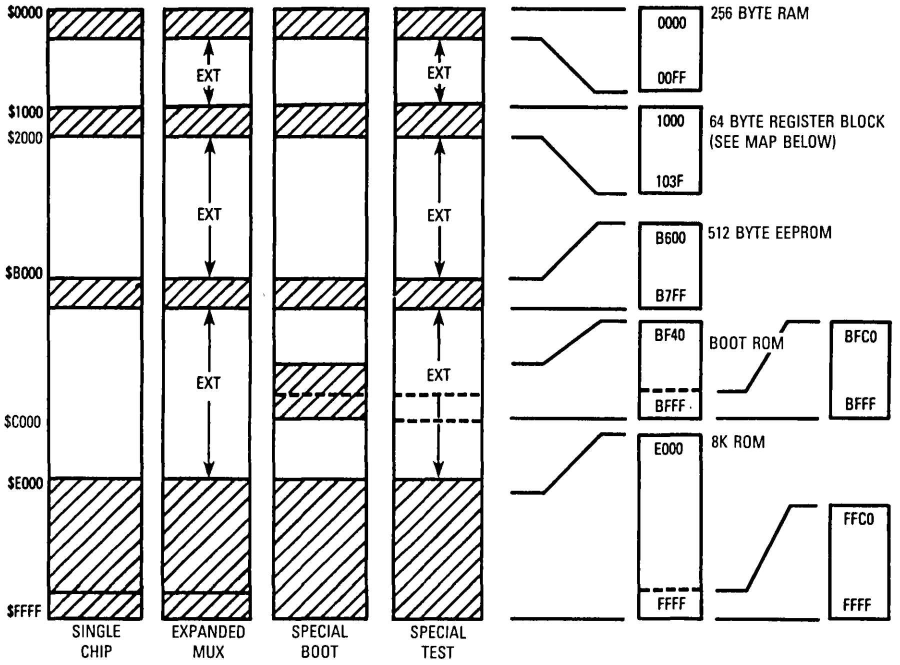
This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.

15	IX	0

INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.

15	IY	0


$0000
$1000
$2000
$B000
$C000
$E000
$FFFF
EXT
EXT
EXT
EXT
EXT
EXT
SINGLE CHIP (MODE 0)
EXPANDED MUX (MODE 1)
SPECIAL BOOT
SPECIAL TEST
0000
00FF
256 BYTE RAM
1000
103F
64 BYTE REGISTER BLOCK (SEE MAP BELOW)
B600
B7FF
512 BYTE EEPROM
BF40
BFFF
BOOT ROM
BFC0
BFFF
E000
FFFF
8K ROM
FFC0
FFFF

NOTES:

1. Either or both the internal RAM and registers can be remapped to any 4K boundard by software.
2. Either or both the ROM and EEPROM can be disabled using a control register (CONFIG) which is implemented with EEPROM cells.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TOC5	Output Compare 5 Register
$101F	Bit 7	—	—	—	—	—	—	Bit 0		
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021			EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Register 1
$1023	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Register 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Register 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Register 2
$1026	DDRA7	PAEN	PAMOD	PEDGE			RTR1	RTR0	PACTL	Pulse Accumulator Control Register
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accumulator Count Register
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1

Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Circuitry
$103B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG	EEPROM Programming Control Register
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Register
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F					NOSEC	NOCOP	ROMON	EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

15	PC	0

STACK POINTER (SP)

The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the accumulators A and B and registers IX and IY can be stored during certain instructions.

15	SP	0

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

7							0
S	X	H	I	N	Z	V	C

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or XIRQ) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

RESETS

The MCU can be reset four ways: 1) an active low input to the RESET pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

RESET PIN

To request an external reset, the RESET pin must be held low for eight E_{cyc} (two E_{cyc} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external RESET pin is low at the end of the power-on delay time, the processor remains in the reset condition until RESET goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the RESET pin low to reset the MCU and the external system.

The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

CLOCK MONITOR RESET

The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the RESET pin is driven low to reset the external system.

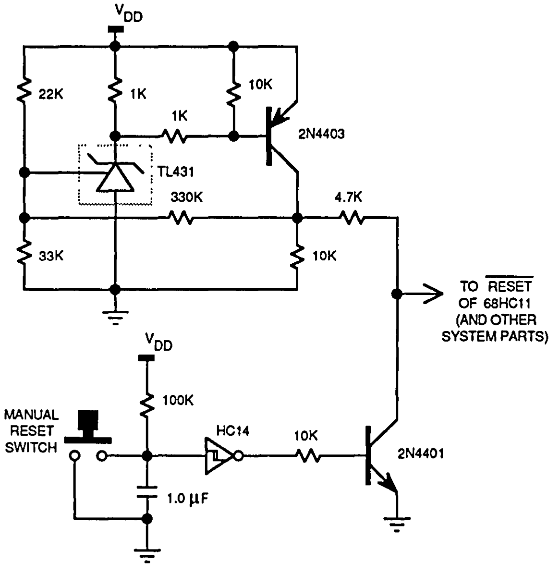
The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.

INTERRUPTS

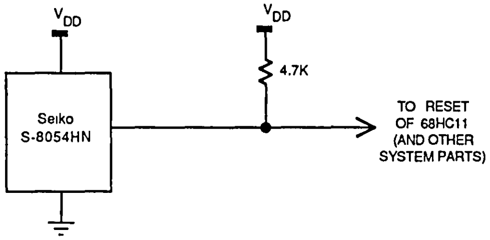
There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the XIRQ

Table 2. COP Timeout Periods

CR1	CR0	$E/2^{15}$ Divided By	XTAL $= 2^{23}$ Timeout $-1/+15.6$ ms	XTAL = 8.0 MHz Timeout $-0/+16.4$ ms	XTAL = 4.9152 MHz Timeout $-0/+26.7$ ms	XTAL = 4.0 MHz Timeout $-0/+32.8$ ms	XTAL = 3.6864 MHz Timeout $-0/+35.6$ ms
0	0	1	15.625 ms	16.384 ms	26 667 ms	32 768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142 22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568 89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz


VDD
22K
1K
10K
1K
2N4403
TL431
330K
4.7K
33K
10K
TO RESET OF 68HC11 (AND OTHER SYSTEM PARTS)
VDD
100K
MANUAL RESET SWITCH
HC14
10K
2N4401
1.0 µF

Reset Circuit with LVI and RC Delay


VDD
VDD
4.7K
Seiko S-8054HN
TO RESET OF 68HC11 (AND OTHER SYSTEM PARTS)

Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the $\overline{\text{XIRQ}}$ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
*	*		
*	*		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Output Compare 5	I Bit	OC5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	$\overline{\text{XIRQ}}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

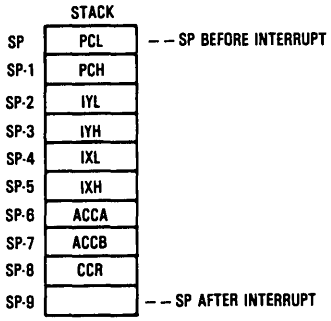


Figure 4. Stacking Order

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be E/2^{13}, E/2^{14}, E/2^{15}, or E/2^{16}.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ or $\overline{\text{RESET}}$. An external interrupt used at

IRQ is only efective if the I bit in the CCR is clear. An external interrupt applied at the XIRQ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the XIRQ request. If the X bit is set, the processing will continue (if no XIRQ interrupt service routine is requested) with the instruction immediately following the STOP instruction. A low input to the RESET pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

If the internal oscillator is being used, a restart delay is required to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the RESET pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit, and the restart delay will be imposed.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes slightly more power than the STOP mode. In the WAIT mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or RESET. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the WAIT mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces WAIT-mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from $FFFF to $0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has three input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are three 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

TIMER CONTROL REGISTER 2 (TCTL2)

	7	6	5	4	3	2	1	0
	0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET	0	0	0	0	0	0	0	0

Bits 7-6 — Not Implemented
These bits always read zero.

EDGxB and EDGxA — Input Capture x Edge Control
These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to $FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one.

TIMER COMPARE FORCE REGISTER (CFORC)

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

	7	6	5	4	3	2	1	0
	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET	0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action
1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set
0 = Has no meaning

Bits 2-0 — Not Implemented
These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M)

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D)

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1)

	7	6	5	4	3	2	1	0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET	0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode

OL2-OL5 — Output Level

These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position.

	7	6	5	4	3	2	1	0
	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F
RESET	0	0	0	0	0	0	0	0

OCxF — Output Compare x Flag

Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).

1 = Bit cleared
0 = Not affected

ICxF — Input Capture x Flag

Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).

1 = Bit cleared
0 = Not affected

TIMER INTERRUPT MASK REGISTER 2 (TMSK2)

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG1. Two timer prescaler bits are also included in this register.

	7	6	5	4	3	2	1	0
	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable

1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled

RTII — RTI Interrupt Enable

1 = Interrupt requested when RTIF = 1
0 = RTIF [illegible]

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2)

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

	7	6	5	4	3	2	1	0
	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

TOF — Timer Overflow

Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag

Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag

Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag

Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented

These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0
	DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0
RESET	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7

1 = Output
0 = Input only

PAEN — Pulse-Accumulator System Enable

1 = Pulse accumulator on
0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode

1 = Gated time accumulator
0 = External even counting

PEDGE — Pulse Accumulator Edge Control

This bit provides clock action along with PAMOD.

1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.
0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

Bits 3-2 — Not Implemented

These bits always read zero.

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

EEPROM PROGRAMMING

The 512 bytes of EEPROM are located at $B600 through $B7FF and have the same read cycle time as the internal ROM. Programming of the EEPROM is controlled by the EEPROM programming control register (PPROG). The EEPROM is disabled when the EEON bit in the system configuration register (CONFIG) is zero. Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz, the efficiency of this charge pump decreases, which increases the time required to program or erase a location. Recommended program and erase time is 10 milliseconds when the E clock is between 2 MHz and should be increased to as much as 20 milliseconds when E clock is between 1 MHz and 2 MHz. When E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. The following paragraphs describe how to program or erase the EEPROM using the PPROG control register.

ERASING THE EEPROM

Erasure of the EEPROM is controled by bit settings in PPROG. Programs can be written to perform bulk, row, or byte erase. In bulk erase, all 512 bytes of the EEPROM are erased. In row erase, 16 bytes ($B600-$B60F, $B610-$B61F), etc) are erased. Other MCU operations can continue to be performed during erasing provided the operations do not include reads of data from EEPROM.

PROGRAMMING EEPROM

During programming, the ROW and BYTE bits are not used. If the E clock frequency is 1 MHz or less, the CSEL bit in the OPTION register must be set. Zeros must be erased by a separate erase operation before programming. Other MCU operations can continue to be performed during programming provided the operations do not include reads of data from EEPROM.

EEPROM PROGRAMMING CONTROL REGISTER (PPROG) $103B

This 8-bit register is used to control programming and erasure of the EEPROM. This register is cleared on reset so the EEPROM is configured for normal reads.

	7	6	5	4	3	2	1	0
	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET	0	0	0	0	0	0	0	0

ODD — Program Odd Rows (TEST)
EVEN — Program Even Rows (TEST)
Bit 5 — Not Implemented
  This bit always reads zero.
BYTE — Byte Erase Select
  This bit overrides the ROW bit.
    1 = Erase only one byte
    0 = Row or bulk erase
ROW — Row Erase Select
  If BYTE bit = 1, ROW has no meaning.
    1 = Row erase
    0 = Bulk or byte erase
ERASE — Erase Mode Select
    1 = Erase mode
    0 = Normal read or program
EELAT — EEPROM Latch Control
    1 = EEPROM Address and data configured for programmming/erasing
    0 = EEPROM Address and data configured for read mode
EEPGM — EEPROM Programming Voltage Enable
    1 = Programming voltage turned on
    0 = Programming voltage turned off

NOTE

If an attempt is made to set both the EELAT and EEPGM bit in the same write cycle, neither will be set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation currently in progress is not disturbed. If no EEPROM address is written between when EECAT is set and EEPGM is set, then no program or erase operation will take place. These safeguards were included to prevent accidental EEPROM changes in cases of program runaway. Mask set A38P, A49N, and date codes before 86xx do not have these safeguards.

ERASING THE CONFIG REGISTER

Erasing the CONFIG register follows the same procedures as that used for the EEPROM except that only bulk erase can be used on the CONFIG register. When the CONFIG register is erased, the EEPROM array is also erased. On mask set B96D, the CONFIG register may only be erased while the MCU is operating in the test or bootstrap mode.The bulk erase restriction on CONFIG is not present on all derivatives in the MC68HC11 family. Please check the applicable data sheet or technical summary for the restrictions.

PROGRAMMING THE CONFIG REGISTER

Programming the CONFIG register follows the same procedures as that used for the EEPROM except the CONFIG register address is used. On mask set B96D, the CONFIG register may only be programmed while the MCU is operating in the test or bootstrap mode.

SYSTEM CONFIGURATION REGISTER (CONFIG)

The CONFIG is implemented in EEPROM cells and controls the presence of ROM and EEPROM in the memory map and enables the COP watchdog system.

7	6	5	4	3	2	1	0
0	0	0	0	NOSEC	NOCOP	ROMON	EEON

Bits 7-4 - Not Implemented
  These bits are always read as zero.
NOSEC — Security Mode Disable Bit
  This bit is only implemented if it is specifically requested with the submission of mask ROM information.
    1 = Disable security mode
    0 = Enable security mode
NOCOP — COP System Disable
    1 = COP watchdog system disabled
    0 = COP watchdog system enabled
ROMON — Enable On-Chip ROM
  When this bit is programmed to "zero", the 8K ROM is disabled, and that memory space becomes externally accessed space. In the single-chip mode, the internal 8K ROM is enabled regardless of the state of the ROMON bit.
EEON — Enable On-Chip EEPROM
  When this bit is programmed to "zero", the 512-byte EEPROM is disabled, and that memory space becomes externally accessed space.

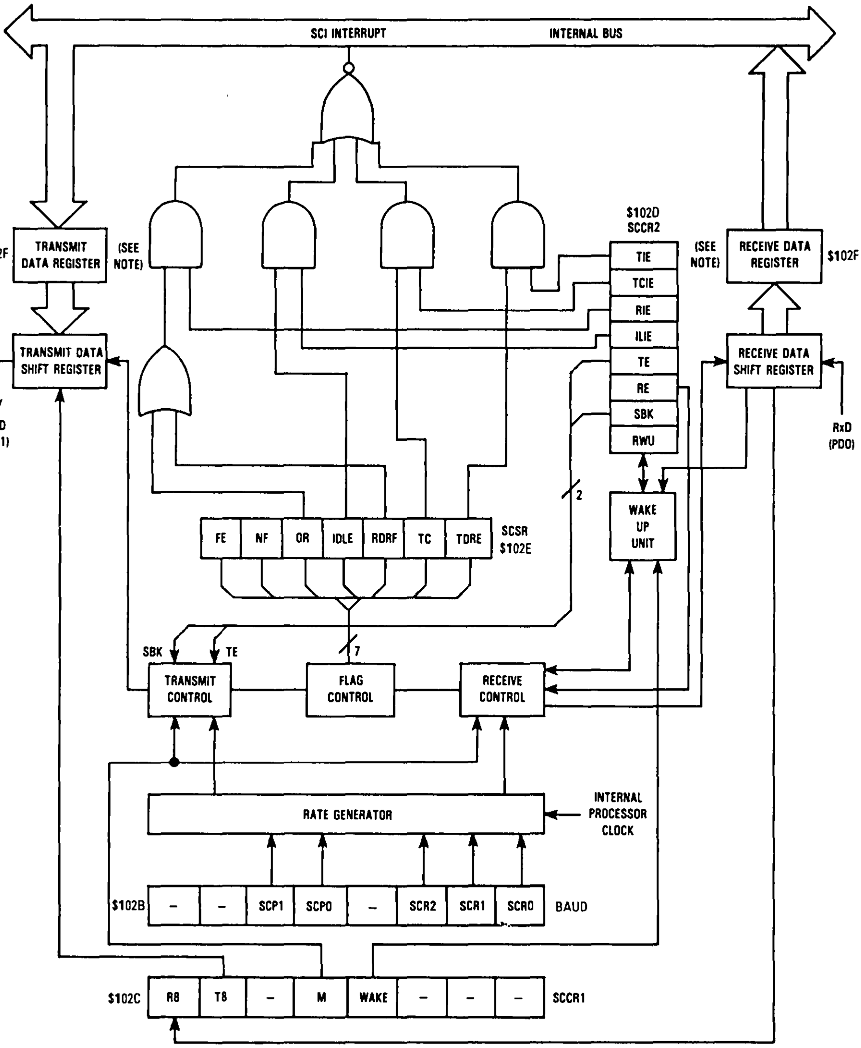
SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins: PD0 for receive data (RxD) and PD1 for the transmit data (TxD). The baud-rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/ reception of a message;


SCI INTERRUPT
INTERNAL BUS
$102F
TRANSMIT DATA REGISTER
(SEE NOTE)
TRANSMIT DATA SHIFT REGISTER
TxD (PD1)
$102D SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
(SEE NOTE)
RECEIVE DATA REGISTER
$102F
RECEIVE DATA SHIFT REGISTER
RxD (PD0)
WAKE UP UNIT
2
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $102E
7
SBK
TE
TRANSMIT CONTROL
FLAG CONTROL
RECEIVE CONTROL
RATE GENERATOR
INTERNAL PROCESSOR CLOCK
$102B
–
–
SCP1
SCP0
–
SCR2
SCR1
SCR0
BAUD
$102C
R8
T8
–
M
WAKE
–
–
–
SCCR1

NOTE: The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read.

Figure 5. SCI Block Diagram

2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR)

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1)

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

	7	6	5	4	3	2	1	0
	R8	T8	0	M	WAKE	0	0	0
RESET	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.

T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.

Bit 5 — Not Implemented
This bit always reads zero.

M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line

Bits 2-0 — Not Implemented
These bits always read zero.

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt if TDRE = 1
0 = TDR interrupts disabled

TCIE — Transmit-Complete Interrupt Enable
1 = SCI interrupt if TC = 1
0 = TC interrupts disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt if RDRF or OR = 1
0 = RDRF or OR interrupt disabled

ILIE — Idle-Line Interrupt Enable
1 = SCI interrupt if IDLE = 1
0 = IDLE interrupts disabled

TE — Transmit Enable
1 = Transmit shift register output is applied to the TxD line
0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
1 = Receiver enabled
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited

RWU — Receiver Wake Up
When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the WAKE bit is zero, RWU is cleared by the SCI logic

after receiving 10 (M=0) or 11 (M=1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If this bit is toggled set and cleared, the transmitter sends 10 (M=0) or 11 (M=1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET	1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty

1=Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register

0=Cleared by a read of SCSR (with TDRE=1) followed by a write to SCDR

TC — Transmit Complete

1=Automatically set when all data frame, preamble, or break condition transmissions are complete

0=Cleared by a read of SCSR (with TC=1) followed by a write to SCDR

RDRF — Receive Data Register Full

1=Automatically set when a character is transferred from the receiver shift register to the SCDR

0=Cleared by a read of SCSR (with RDRF=1) followed by a read of SCDR

IDLE — Idle-Line Detect

This bit is inhibited while RWU=1.

1=Automatically set when the receiver serial input becomes idle after having been active

0=Cleared by a read of SCSR (with IDLE=1) followed by a read of SCDR

OR — Overrun Error

1=Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read

0=Cleared by a read of SCSR (with OR=1) followed by a read of SCDR

NF — Noise Flag

1=Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame

0=Cleared by a read of SCSR (with NF=1) followed by a write to SCDR

FE — Framing Error

1=Automatically set when a logic 0 is detected where a stop bit was expected

0=Cleared by a read of SCSR (with FE=1) followed by a read of SCDR

Bit 0 — Not Implemented

This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

	7	6	5	4	3	2	1	0
	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET	0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)

This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.

Bit 6 — Not Implemented

This bit always reads zero.

SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects

These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)

This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

SCR2-SCR0 — SCI Baud-Rate Selects

These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

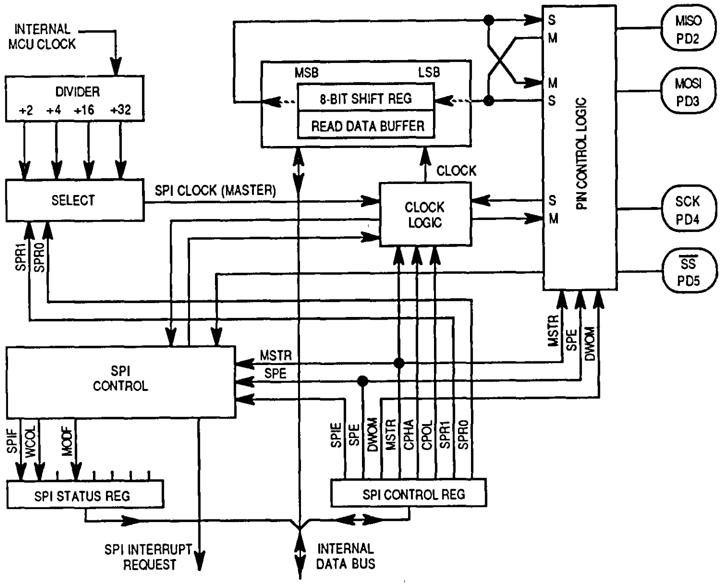
SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	**1**	**0**		**131.072 K Baud**	**32.768 K Baud**	**76.80 K Baud**	**19.20 K Baud**	**9600 Baud**
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.



INTERNAL MCU CLOCK
DIVIDER
÷2 ÷4 ÷16 ÷32
SELECT
SPR1
SPR0
SPI CLOCK (MASTER)
MSB
LSB
8-BIT SHIFT REG
READ DATA BUFFER
CLOCK
CLOCK LOGIC
S
M
M
S
S
M
PIN CONTROL LOGIC
MISO PD2
MOSI PD3
SCK PD4
$\overline{SS}$ PD5
MSTR
SPE
DWOM
SPI CONTROL
MSTR
SPE
SPIF
WCOL
MODF
SPI STATUS REG
SPIE
SPE
DWOM
MSTR
CPHA
CPOL
SPR1
SPR0
SPI CONTROL REG
SPI INTERRUPT REQUEST
INTERNAL DATA BUS

Figure 6. SPI Block Diagram

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt if SPIF = 1
0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

DWOM — Port D Wire-OR Mode Option
This bit affects all six port D pins together.
1 = Port D outputs act as open-drain outputs
0 = Port D outputs are normal CMOS outputs

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
This bit selects the polarity of the SCK clock.
1 = SCK line idles high
0 = SCK line idles low

CPHA — Clock Phase
This bit selects one of two fundamentally different clock protocols. Refer to Figure 7.

SPR1 and SPR0 — SPI Clock Rate Select
These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR)

	7	6	5	4	3	2	1	0
	SPIF	WCOL	0	MODF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag
1 = Automatically set when data transfer is complete between processor and external device
0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision
If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{SS}$ is low and ends when the SPIF flag gets set.
1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred
0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented
This bit always reads zero.

MODF — Mode Fault
This bit indicates the possibility of a multi-master conflict for system control and therefore allows a

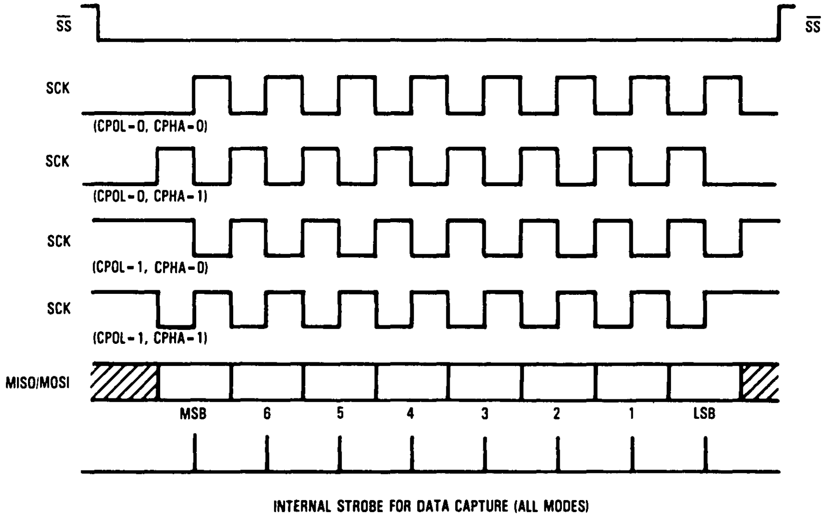


Figure 7. Data Clock Timing Diagram

proper exit from system operation to a reset or default system state.

1 = Automatically set when a master device has its $\overline{SS}$ pin pulled low

0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented

These bits always read zero.

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.
2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

NOTE

In the 48-pin dual-in-line package, four conversion channels are not implemented. These include channels four through seven.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16×16	FDIV
Integer Divide 16×16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8×8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if = zero	BEQ
Branch if≥zero	BGE
Branch if)zero	BGT
Branch if Higher	BHI
Branch if≤Zero	BLE
Branch if Lower or Same	BLS
Branch if(Zero	BLT
Branch if Minus	BMI
Branch if not = Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

— Continued —

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

OPCODE MAP SUMMARY

Table 4 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

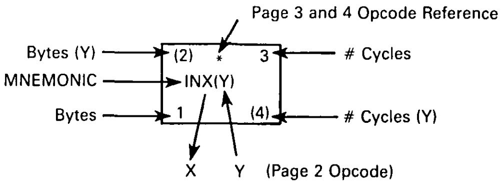
Table 6. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
LOW HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI LOW
0 0000	TEST 1	SBA 2 1	BRA 3 2	(2) TSX(Y) 3 1 (4)	NEGA 2 1	NEGB 2 1	(3) NEG 6 2 (7)	NEG 6 3	SUBA 2 2	SUBA 3 2	(3) SUBA 4 2 (5)	SUBA 4 3	SUBB 2 2	SUBB 3 2	(3) SUBB 4 2 (5)	SUBB 4 3	0
1 0001	NOP 2 1	CBA 2 1	BRN 3 2	INS 3 1					CMPA 2 2	CMPA 3 2	(3) CMPA 4 2 (5)	CMPA 4 3	CMPB 2 2	CMPB 3 2	(3) CMPB 4 2 (5)	CMPB 4 3	1
2 0010	IDIV 41 1	BRSET 6 4	BHI 3 2	PULA 4 1					SBCA 2 2	SBCA 3 2	(3) SBCA 4 2 (5)	SBCA 4 3	SBCB 2 2	SBCB 3 2	(3) SBCB 4 2 (5)	SBCB 4 3	2
3 0011	FDIV 41 1	BRCLR 6 4	BLS 3 2	PULB 4 1	COMA 2 1	COMB 2 1	(3) COM 6 2 (7)	COM 6 3	* SUBD 4 3	* SUBD 5 2	(3) * SUBD 6 2 (7)	* SUBD 6 3	ADDD 4 3	ADDD 5 2	(3) ADDD 6 2 (7)	ADDD 6 3	3
4 0100	LSRD 3 1	BSET 6 3	(BHS) BCC 3 2	DES 3 1	LSRA 2 1	LSRB 2 1	(3) LSR 6 2 (7)	LSR 6 3	ANDA 2 2	ANDA 3 2	(3) ANDA 4 2 (5)	ANDA 4 3	ANDB 2 2	ANDB 3 2	(3) ANDB 4 2 (5)	ANDB 4 3	4
5 0101	(LSLD) ASLD 3 1	BCLR 6 3	(BLO) BCS 3 2	(2) TX(Y)S 3 1 (4)					BITA 2 2	BITA 3 2	(3) BITA 4 2 (5)	BITA 4 3	BITB 2 2	BITB 3 2	(3) BITB 4 2 (5)	BITB 4 3	5
6 0110	TAP 2 1	TAB 2 1	BNE 3 2	PSHA 3 1	RORA 2 1	RORB 2 1	(3) ROR 6 2 (7)	ROR 6 3	LDAA 2 2	LDAA 3 2	(3) LDAA 4 2 (5)	LDAA 4 3	LDBB 2 2	LDBB 3 2	(3) LDBB 4 2 (5)	LDBB 4 3	6
7 0111	TPA 2 1	TBA 2 1	BEQ 3 2	PSHB 3 1	ASRA 2 1	ASRB 2 1	(3) ASR 6 2 (7)	ASR 6 3		STAA 3 2	(3) STAA 4 2 (5)	STAA 4 3		STBB 3 2	(3) STBB 4 2 (5)	STBB 4 3	7
8 1000	(2) INX(Y) 3 1 (4)	PAGE 2	BVC 3 2	(2) PULX(Y) 5 1 (6)	ASLA 2 1	ASLB 2 1	(3) ASL 6 2 (7)	ASL 6 3	EORA 2 2	EORA 3 2	(3) EORA 4 2 (5)	EORA 4 3	EORB 2 2	EORB 3 2	(3) EORB 4 2 (5)	EORB 4 3	8
9 1001	(2) DEX(Y) 3 1 (4)	DAA 2 1	BVS 3 2	RTS 5 1	ROLA 2 1	ROLB 2 1	(3) ROL 6 2 (7)	ROL 6 3	ADCA 2 2	ADCA 3 2	(3) ADCA 4 2 (5)	ADCA 4 3	ADCB 2 2	ADCB 3 2	(3) ADCB 4 2 (5)	ADCB 4 3	9
A 1010	CLV 2 1	PAGE 3	BPL 3 2	(2) ABX(Y) 3 1 (4)	DECA 2 1	DECB 2 1	(3) DEC 6 2 (7)	DEC 6 3	ORAA 2 2	ORAA 3 2	(3) ORAA 4 2 (5)	ORAA 4 3	ORAB 2 2	ORAB 3 2	(3) ORAB 4 2 (5)	ORAB 4 3	A
B 1011	SEV 2 1	ABA 2 1	BMI 3 2	RTI 12 1					ADDA 2 2	ADDA 3 2	(3) ADDA 4 2 (5)	ADDA 4 3	ADDB 2 2	ADDB 3 2	(3) ADDB 4 2 (5)	ADDB 4 3	B
C 1100	CLC 2 1	(4) BSET 7 3 (8)	BGE 3 2	(2) PSHX(Y) 4 1 (5)	INCA 2 1	INCB 2 1	(3) INC 6 2 (7)	INC 6 3	(4) CPX(Y) 4 3 5	(3) CPX(Y) 5 2 (6)	(3) * CPX(Y) 6 2 (7)	(4) CPX(Y) 6 3 (7)	LDD 3 3	LDD 4 2	(3) LDD 5 2 (6)	LDD 5 3	C
D 1101	SEC 2 1	(4) BCLR 7 3 (8)	BLT 3 2	MUL 10 1	TSTA 2 1	TSTB 2 1	(3) TST 6 2 (7)	TST 6 3	BSR 6 2	JSR 5 2	(3) JSR 6 2 (7)	JSR 6 3	PAGE 4	STD 4 2	(3) STD 5 2 (6)	STD 5 3	D
E 1110	CLI 2 1	(5) BRSET 7 4 (8)	BGT 3 2	WAI 12 1			(3) JMP 3 2 (4)	JMP 3 3	LDS 3 3	LDS 4 2	(3) LDS 5 2 (6)	LDS 5 3	(4) LDX(Y) 3 3 (4)	(3) LDX(Y) 4 2 (5)	(3) LDX(Y) 5 2 (6)	(4) * LDX(Y) 5 3 (6)	E
F 1111	SEI 2 1	(5) BRCLR 7 4 (8)	BLE 3 2	SWI 14 1	CLRA 2 1	CLRB 2 1	(3) CLR 6 2 (7)	CLR 6 3	(2) XGDX(Y) 3 1 (4)	STS 4 2	(3) STS 5 2 (6)	STS 5 3	STOP 2 1	(3) STX(Y) 4 2 (5)	(3) STX(Y) 5 2 (6)	(4) * STX(Y) 5 3 (6)	F

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

***Page 3 and 4 Opcode Reference**

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC11A8 MC68HC11A8V MC68HC11A8M	T_A	T_L to T_H −40 to 85 −40 to 105 −40 to 125	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC) Plastic 48-Pin Dual-In-Line	θ_{JA}	 50 40	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

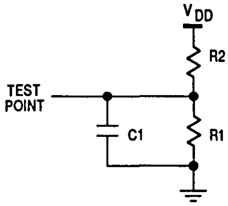
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD}=5.0 Vdc±10%, V_{SS}=0 Vdc, T_A=T_L to T_H, unless otherwise noted)

Characteristic		Symbol	Min	Max	Unit
Output Voltage	All Outputs	V_{OL}	—	0.1	V
I_{Load} = ±10.0 μA (see Note 1)	All Outputs Except $\overline{\text{RESET}}$ and MODA	V_{OH}	$V_{DD}-0.1$	—	
Output High Voltage I_{Load} = −0.8 mA, V_{DD}=4.5 V (see Note 1)	All Outputs Except $\overline{\text{RESET}}$, XTAL, and MODA	V_{OH}	$V_{DD}-0.8$	—	V
Output Low Voltage I_{Load}=1.6 mA	All Outputs Except XTAL	V_{OL}	—	0.4	V
Input High Voltage	All Inputs Except $\overline{\text{RESET}}$	V_{IH}	$0.7\times V_{DD}$	V_{DD}	V
	$\overline{\text{RESET}}$		$0.8\times V_{DD}$	V_{DD}	
Input Low Voltage	All Inputs	V_{IL}	V_{SS}	$0.2\times V_{DD}$	V
I/O Ports, Three-State Leakage V_{in}=V_{IH} or V_{IL}	PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	I_{OZ}	—	±10	μA
Input Current (see Note 2)		I_{in}			μA
V_{in}=V_{DD} or V_{SS}	PA0-PA2, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$		—	±1	
V_{in}=V_{DD} or V_{SS}	MODB/V_{STBY}		—	±10	
RAM Standby Voltage	Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	μA
Total Supply Current (see Note 3)					
RUN:		I_{DD}			
Single Chip			—	15	mA
Expanded Multiplexed			—	27	mA
WAIT:		W_{IDD}			
All Peripheral Functions Shut Down					
Single-Chip Mode			—	6	mA
Expanded Multiplexed Mode			—	10	mA
STOP:		S_{IDD}			
No Clocks, Single-Chip Mode			—	100	μA
Input Capacitance	PA0-PA2, PE0-PE7, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL	C_{in}	—	8	pF
	PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$		—	12	
Power Dissipation	Single-Chip Mode	P_D	—	85	mW
	Expanded-Multiplexed Mode		—	150	

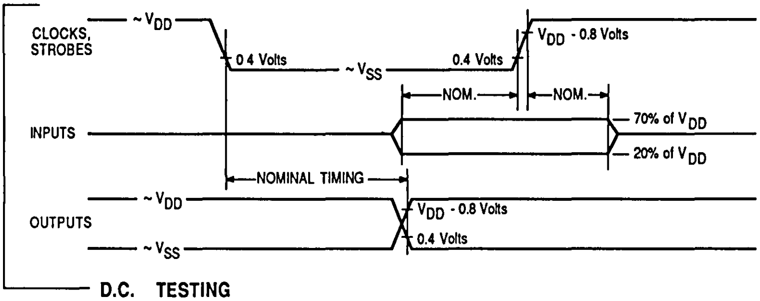
NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL}\leq 0.2$ V,
   $V_{IH}\geq V_{DD}-0.2$ V,
   No dc loads,
   EXTAL is driven with a square wave, and
   t_{cyc}=476.5 ns.

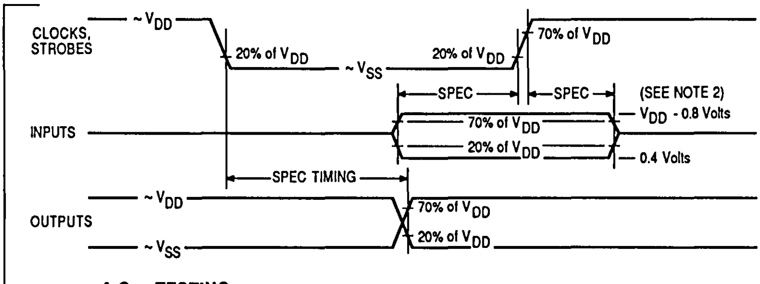

VDD
R2
TEST POINT
C1
R1

Equivalent Test Load[1]

Pins	R1	R2	C1
PA3-PA7 PB0-PB7 PC0-PC7 PD0, PD5 E, AS, R/W	3.26K	2.38K	90pF
PD1-PD4	3.26K	2.38K	200pF


CLOCKS, STROBES
~VDD
0.4 Volts
~VSS
0.4 Volts
VDD - 0.8 Volts
NOM.
NOM.
INPUTS
70% of VDD
20% of VDD
NOMINAL TIMING
OUTPUTS
~VDD
~VSS
VDD - 0.8 Volts
0.4 Volts

D.C. TESTING


CLOCKS, STROBES
~VDD
20% of VDD
~VSS
20% of VDD
70% of VDD
SPEC
SPEC
(SEE NOTE 2)
INPUTS
70% of VDD
20% of VDD
VDD - 0.8 Volts
0.4 Volts
SPEC TIMING
OUTPUTS
~VDD
~VSS
70% of VDD
20% of VDD

A.C. TESTING

NOTES:

1. Full test loads are applied during all ac electrical test and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and $V_{DD} - 0.8$ volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

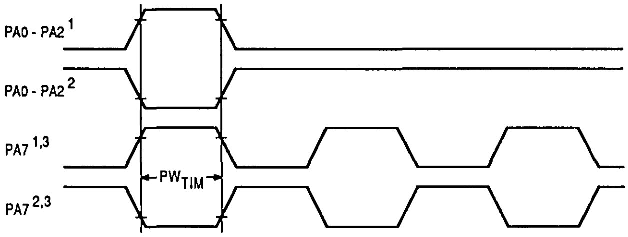
Figure 8. Test Methods

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and12) t_{PCS} = 1/4 t_{cyc} − 50 ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time; May be Preempted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) PW_{IRQ} = t_{cyc} + 20 ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) PW_{TIM} = t_{cyc} + 20 ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:

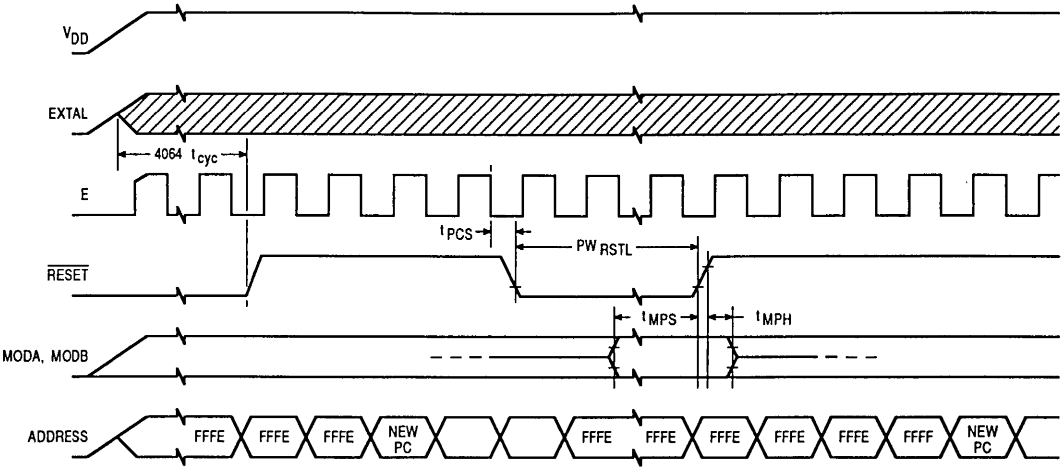
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



NOTES:

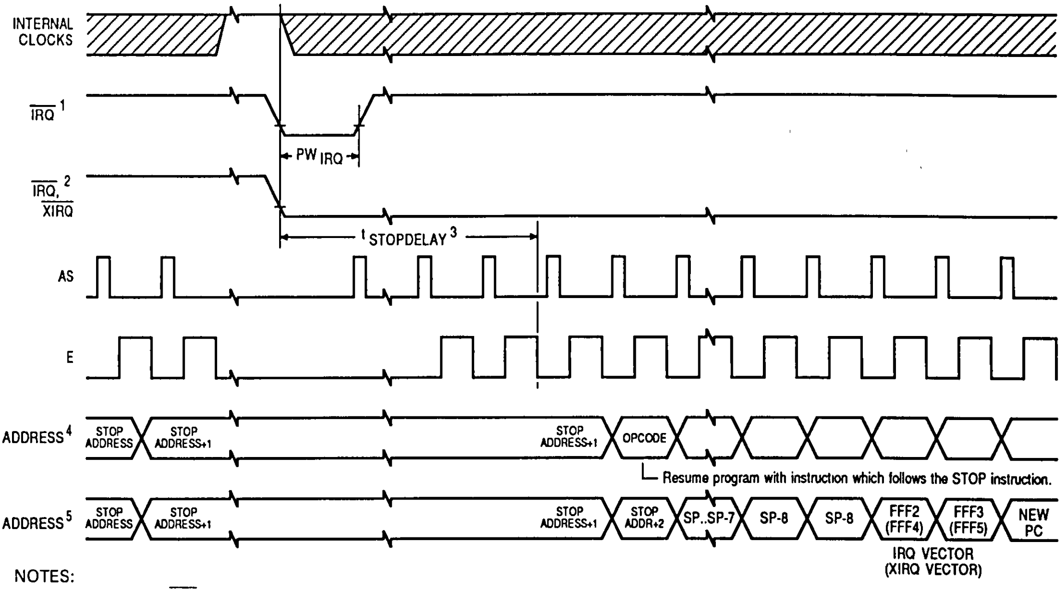
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 9. Timer Inputs Timing Diagram


VDD
EXTAL
4064 tcyc
E
tPCS
PW RSTL
RESET
tMPS
tMPH
MODA, MODB
ADDRESS
FFFE
FFFE
FFFE
NEW PC
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
NEW PC

NOTE Refer to Table 9-7 for pin states during $\overline{\text{RESET}}$

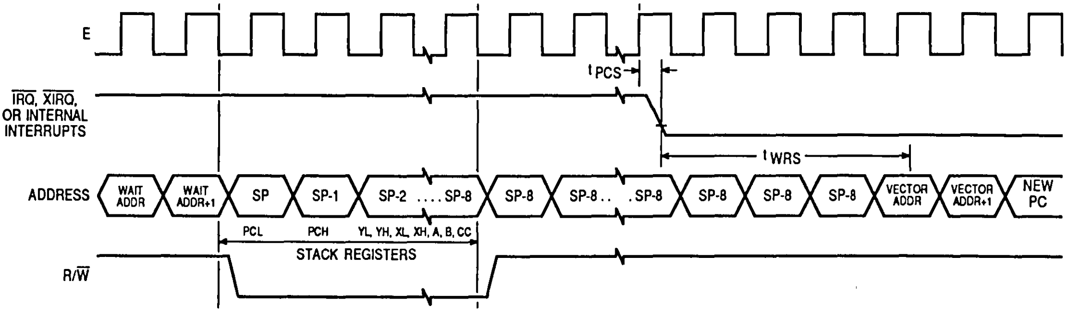
Figure 10. POR External Reset Timing Diagram


INTERNAL CLOCKS
IRQ 1
PW IRQ
IRQ, 2 XIRQ
t STOPDELAY 3
AS
E
ADDRESS 4
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
OPCODE
Resume program with instruction which follows the STOP instruction.
ADDRESS 5
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
STOP ADDR+2
SP..SP-7
SP-8
SP-8
FFF2 (FFF4)
FFF3 (FFF5)
NEW PC
IRQ VECTOR (XIRQ VECTOR)

NOTES:

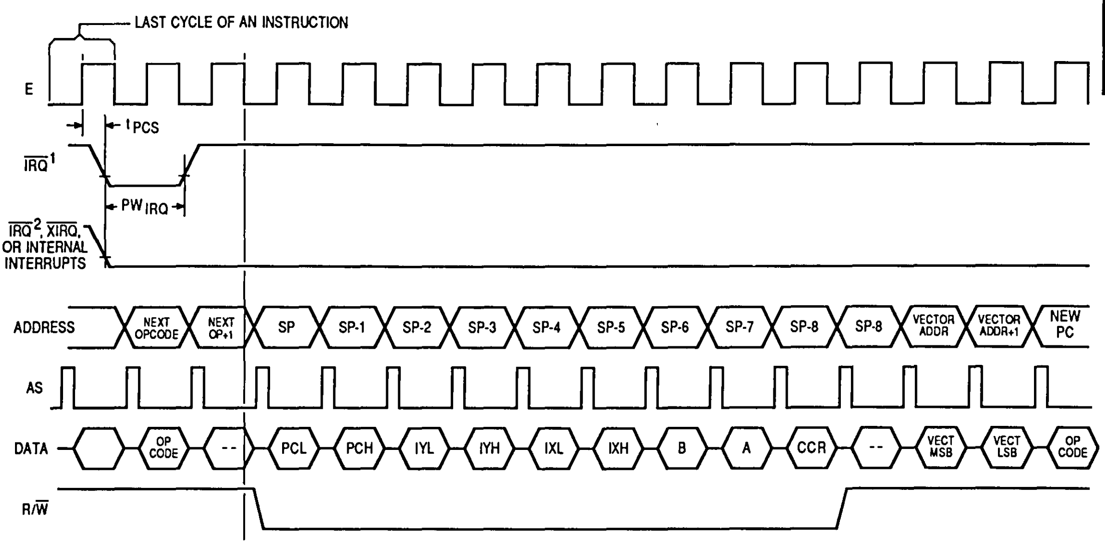
1. Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 0)
3. $t_{\text{STOPDELAY}}$ = 4064 t_{cyc} if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{\text{XIQ}}$ with X bit in CCR = 1.
5. $\overline{\text{IRQ}}$ or ($\overline{\text{XIRQ}}$ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram


E
tPCS
IRQ, XIRQ, OR INTERNAL INTERRUPTS
tWRS
ADDRESS
WAIT ADDR
WAIT ADDR+1
SP
SP-1
SP-2 .... SP-8
SP-8
SP-8.. .SP-8
SP-8
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
PCL
PCH
YL, YH, XL, XH, A, B, CC
STACK REGISTERS
R/W

NOTES.
1. Refer to Table 9-7 for pin states during WAIT
2. RESET will also cause recovery from WAIT

Figure 12. WAIT Recovery from Interrupt Timing Diagram


LAST CYCLE OF AN INSTRUCTION
E
tPCS
IRQ1
PWIRQ
IRQ2, XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
NEXT OPCODE
NEXT OP+1
SP
SP-1
SP-2
SP-3
SP-4
SP-5
SP-6
SP-7
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
AS
DATA
OP CODE
--
PCL
PCH
IYL
IYH
IXL
IXH
B
A
CCR
--
VECT MSB
VECT LSB
OP CODE
R/W

NOTES:
1. Edge sensitive IRQ pin (IRQE bit = 1)
2. Level sensitive IRQ pin (IRQE bit = 0).

Figure 13. Interrupt Timing Diagram

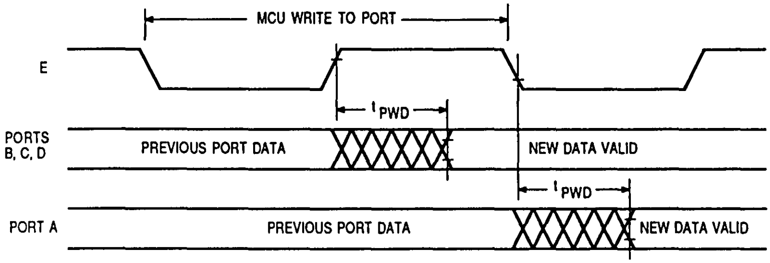
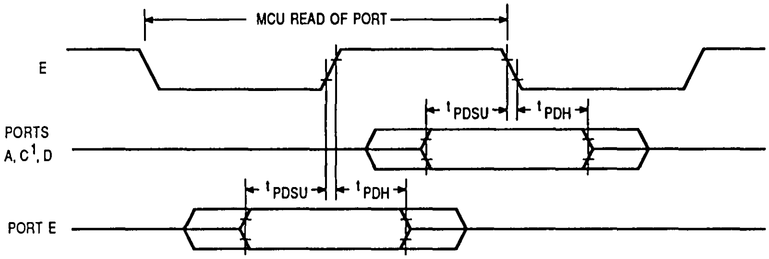

MCU WRITE TO PORT
E
tPWD
PORTS B, C, D
PREVIOUS PORT DATA
NEW DATA VALID
tPWD
PORT A
PREVIOUS PORT DATA
NEW DATA VALID

Figure 14. Port Write Timing Diagram


MCU READ OF PORT
E
tPDSU
tPDH
PORTS A, C1, D
tPDSU
tPDH
PORT E

NOTE1. For non-latched operation of Port C.

Figure 15. Port Read Timing Diagram

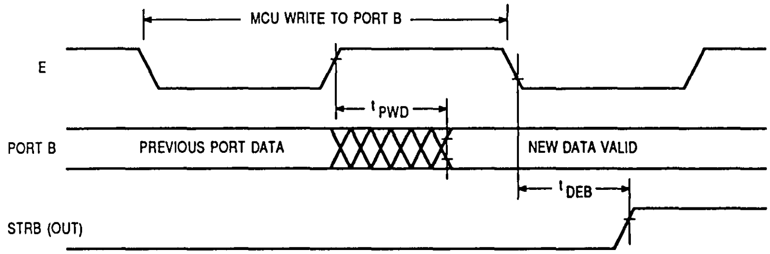

MCU WRITE TO PORT B
E
tPWD
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
tDEB
STRB (OUT)

Figure 16. Simple Output Strobe Timing Diagram

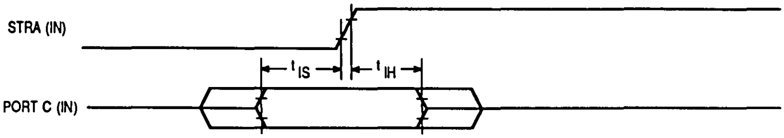
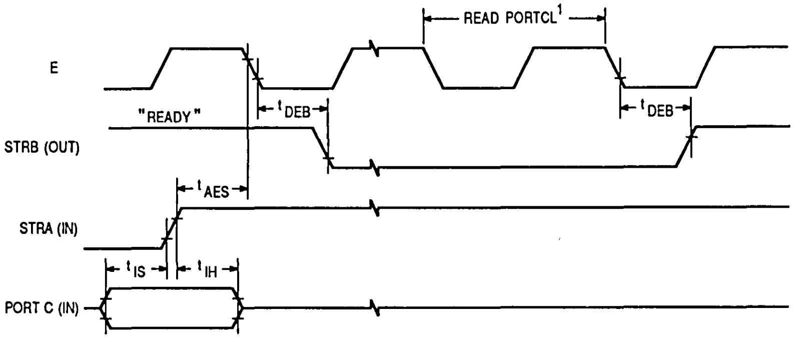
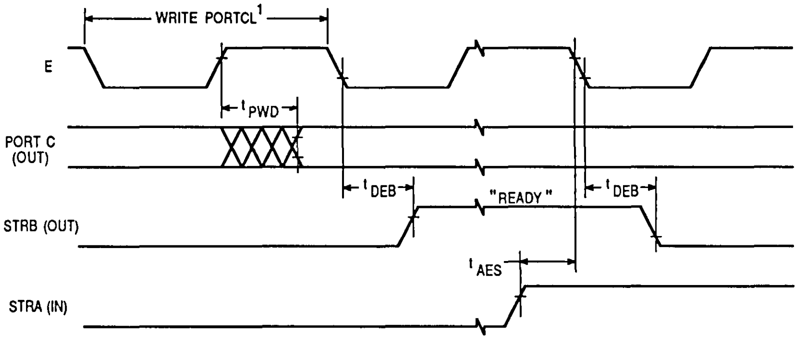

STRA (IN)
tIS
tIH
PORT C (IN)

Figure 17. Simple Input Strobe Timing Diagram


READ PORTCL1
E
tDEB
tDEB
"READY"
STRB (OUT)
tAES
STRA (IN)
tIS
tIH
PORT C (IN)

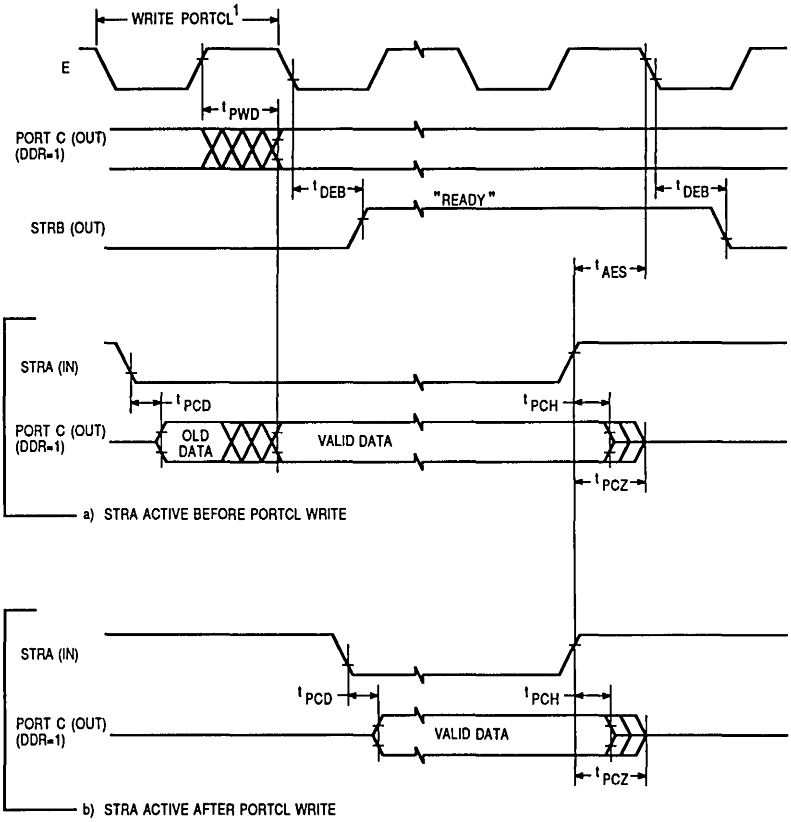
NOTES.
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 18. Port C Input Handshake Timing Diagram


WRITE PORTCL1
E
tPWD
PORT C (OUT)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)

NOTES:
1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 19. Port C Output Handshake Timing Diagram


WRITE PORTCL¹
E
tPWD
PORT C (OUT) (DDR=1)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
OLD DATA
VALID DATA
tPCZ
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
VALID DATA
tPCZ
b) STRA ACTIVE AFTER PORTCL WRITE

NOTES:
1. After reading PIOC with STAF set.
2 Figure shows rising edge STRA (EGA=1) and high true STRB (INVB=1)

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 14)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 15, 17, and 18)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D $t_{PWD} = 1/4\ t_{cyc} + 90$ ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 16 and 17)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 16 and 17)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB $t_{DEB} = 1/4\ t_{cyc} + 100$ ns (See Figure 15, 17, 18, and 19)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 17, 18, 19)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 19)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 19)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 19)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:
1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

A/D CONVERTER CHARACTERISTICS ($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	$V_{DD}+0.1$	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	$V_{SS}-0.1$	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion. a. E Clock b Internal RC Oscillator	 — —	 32 —	 — $t_{cyc}+32$	 t_{cyc} µs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in}=V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in}=V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time: a. E Clock b. Internal RC Oscillator	 — —	 12 —	 — 12	 t_{cyc} µs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7 V_{RL}, V_{RH}	— —	— —	400 1.0	nA µA

NOTES:

1. Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at $\Delta V_R=5$ V ± 10%.

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns) see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns) see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES:

1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of $1/8\ t_{cyc}$ in the above formulas where applicable:
   (a) $(1\text{-DC}) \times 1/4\ t_{cyc}$
   (b) $\text{DC} \times 1/4\ t_{cyc}$
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD}=5.0 Vdc ± 10%, V_{SS}=0 Vdc, $T_A=T_L$ to T_H, see Figure 22)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid (After Enable Edge)**	tv(s)	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	tho	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L=200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{rm} trs	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L=200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{fm} tfs	 — —	 100 2.0	 ns µs

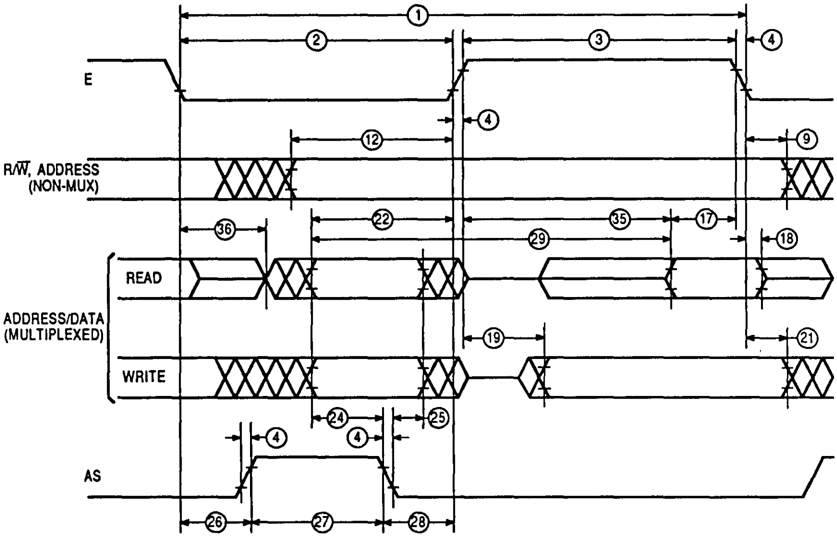
*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.
NOTE:
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

EEPROM CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic		Temperature Range			Unit
		−40 to 85°C	−40 to 105°C	−40 to 125°C	
Programming Time (see Note 1)	Under 1.0 MHz with RC Oscillator Enabled	10	15	20	ms
	1.0 to 2.0 MHz with RC Oscillator Disabled	20	Must Use RC	Must Use RC	
	2.0 MHz (or Anytime RC Oscillator Enabled)	10	15	20	
Erase Time (see Note 1)	Byte, Row, and Bulk	10	10	10	ms
Write/Erase Endurance (see Note 2)		10,000	10,000	10,000	Cycles
Data Retention (see Note 2)		10	10	10	Years

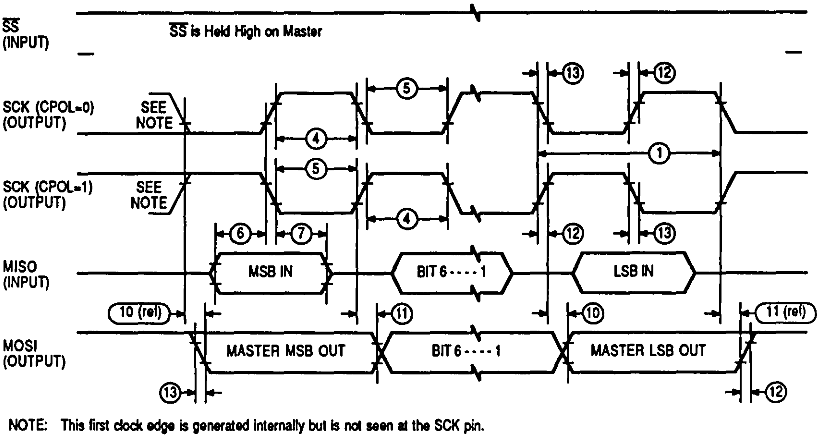
NOTES:

1. The RC oscillator must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. See current quarterly Reliability Monitor report for current failure rate information.

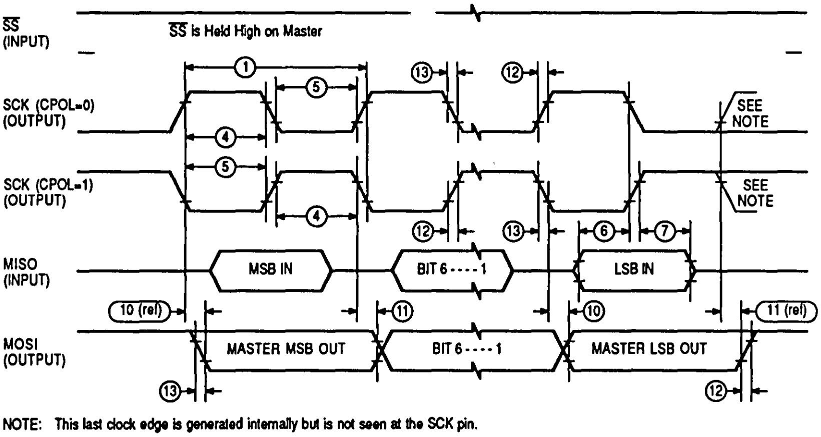


NOTE: Measurement points shown are 20% and 70% V_{DD}

Figure 21. Expansion Bus Timing Diagram

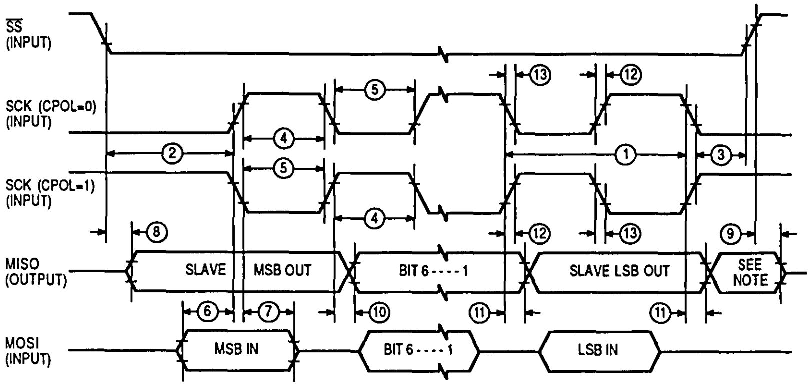

SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
① ④ ⑤ ⑥ ⑦ ⑩ ⑪ ⑫ ⑬
NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SCK (CPOL=1) (OUTPUT)
SEE NOTE
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
10 (ref)
11 (ref)
① ④ ⑤ ⑥ ⑦ ⑩ ⑪ ⑫ ⑬
NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

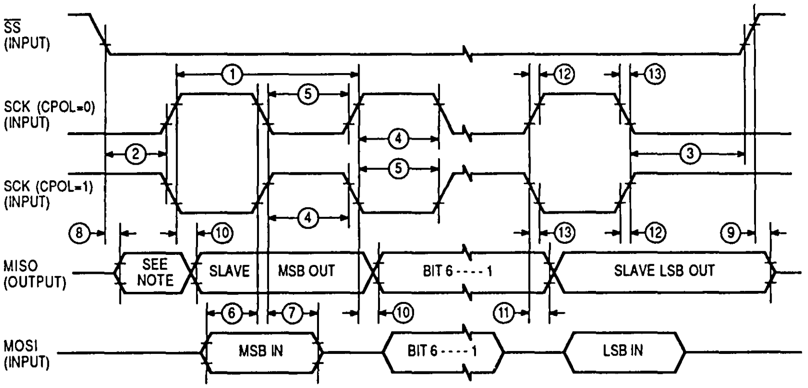
b) SPI MASTER TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6----1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6----1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE: Not defined but normally MSB of character just received.

c) SPI SLAVE TIMING (CPHA=0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6----1
SLAVE LSB OUT
MSB IN
BIT 6----1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE· Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MS-DOS/PC-DOS disk file (360K)
EPROM(s): two 2532/2732 or one 2764

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field-service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MS-DOS™/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

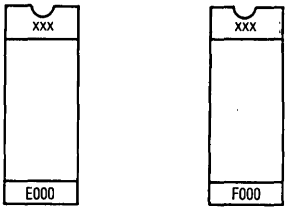
In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5 1/4-inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M68HC11 cross assemblers and linkers on IBM PC-style machines.

EPROMs

Two 2532/2732 or one 2764 type EPROM(s), programmed with the customer's progrm (positive logic sense for address and data), may be submitted for pattern generation. EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure 23 illustrates the markings for the two 2532/2732 EPROMs required to contain the customer program.

All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Figure 23. EPROM Marking

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program ***customer supplied*** blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum-order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

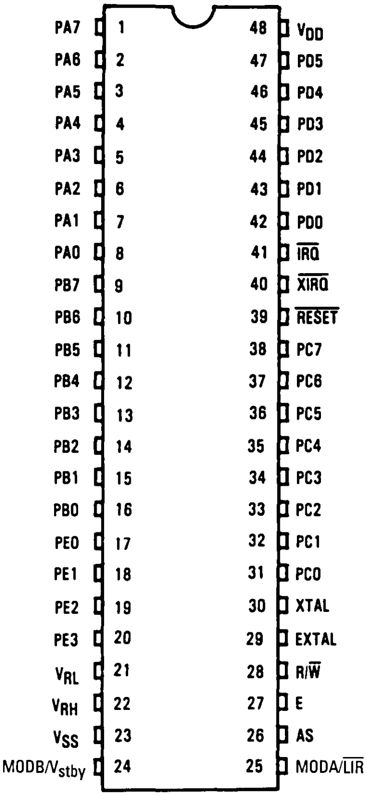
The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC11A8 HCMOS single-chip microcontroller devices.

Package Type	Temperature	CONF	Description	MC Part Number
Plastic (P Suffix)	−40° to +85°C	$0F	BUFFALO ROM	MC68HC11A8P1
PLCC (FN Suffix)	−40° to +85°C	$0F	BUFFALO ROM	MC68HC11A8FN1

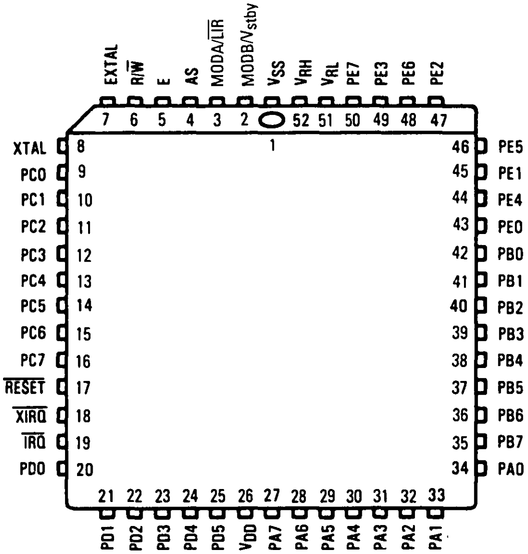
MS-™DOS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

48-Pin Dual-in-Line Package


PA7 1
PA6 2
PA5 3
PA4 4
PA3 5
PA2 6
PA1 7
PA0 8
PB7 9
PB6 10
PB5 11
PB4 12
PB3 13
PB2 14
PB1 15
PB0 16
PE0 17
PE1 18
PE2 19
PE3 20
V_{RL} 21
V_{RH} 22
V_{SS} 23
MODB/V_{stby} 24
48 V_{DD}
47 PD5
46 PD4
45 PD3
44 PD2
43 PD1
42 PD0
41 $\overline{IRQ}$
40 $\overline{XIRQ}$
39 $\overline{RESET}$
38 PC7
37 PC6
36 PC5
35 PC4
34 PC3
33 PC2
32 PC1
31 PC0
30 XTAL
29 EXTAL
28 R/$\overline{W}$
27 E
26 AS
25 MODA/$\overline{LIR}$

52-Lead Quad Package


EXTAL 7
R/$\overline{W}$ 6
E 5
AS 4
MODA/$\overline{LIR}$ 3
MODB/V_{stby} 2
1
V_{SS} 52
V_{RH} 51
V_{RL} 50
PE7 49
PE3 48
PE6 47
PE2
XTAL 8
PC0 9
PC1 10
PC2 11
PC3 12
PC4 13
PC5 14
PC6 15
PC7 16
$\overline{RESET}$ 17
$\overline{XIRQ}$ 18
$\overline{IRQ}$ 19
PD0 20
21 PD1
22 PD2
23 PD3
24 PD4
25 PD5
26 V_{DD}
27 PA7
28 PA6
29 PA5
30 PA4
31 PA3
32 PA2
33 PA1
46 PE5
45 PE1
44 PE4
43 PE0
42 PB0
41 PB1
40 PB2
39 PB3
38 PB4
37 PB5
36 PB6
35 PB7
34 PA0

MC68HC11D3

Product Preview

8-Bit Microcontroller

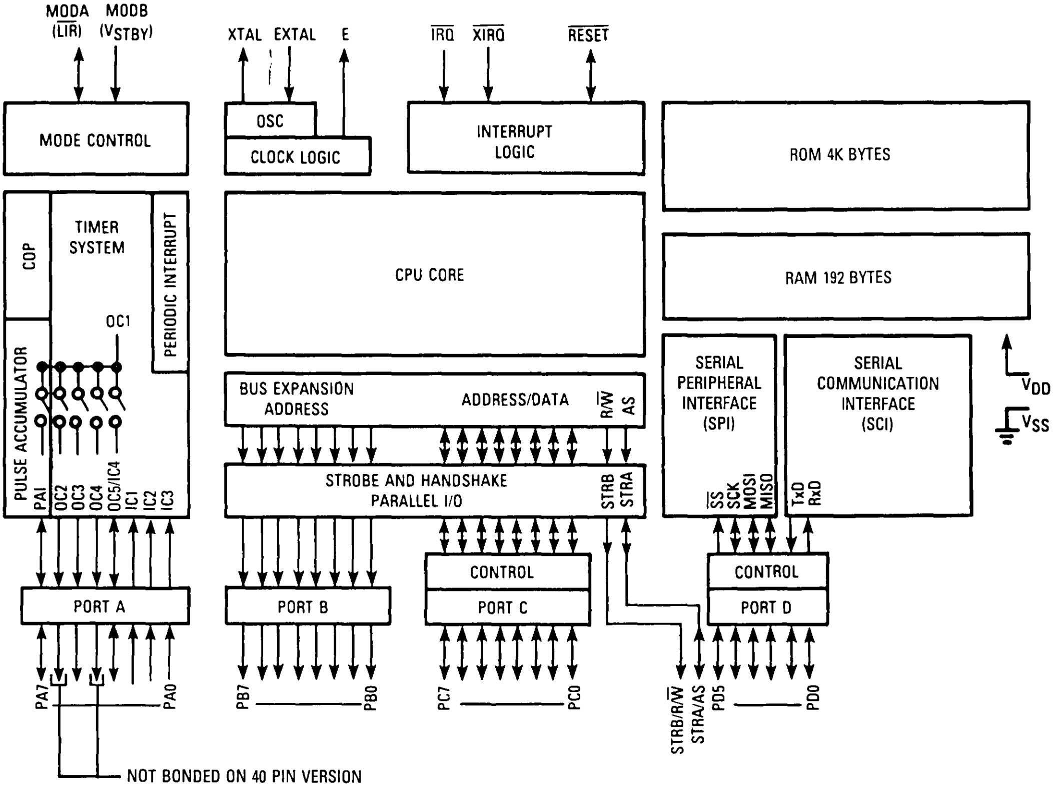
The MC68HC11D3 Microcontroller (MCU) device is similar to the MC68HC11A8. Some exceptions are the 4096 bytes of ROM, 192 bytes of RAM, and no A/D converter (refer to block diagram).

The following are some of the hardware and software features of the MC68HC11D3.

- HCMOS Technology
- 4096 Bytes of User ROM
- 192 Bytes of RAM
- Enhanced 16-Bit Timer System:
  - Four Stage Programmable Prescaler
  - Three or Four Input Captures, Four or Five Output Compares (44 Pin Package)
  - Three or Four Input Captures, One or Two Output Compares (40 Pin Package)
- 8-Bit Pulse Accumulator Circuit
- Enhanced NRZ Serial Communications Interface (SCI)
- Serial Peripheral Interface (SPI)
- Real Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog Timer
- Enhanced M6800/M6801 Instruction Set
- 16 × 16 Integer and Fractional Divide Instruction
- Bit Manipulation
- WAIT and STOP Modes
- Expansion Mode Addressable to 64K
- 40-Pin DIP, 44-Pin PLCC Package

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

BLOCK DIAGRAM


MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
ROM 4K BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
CPU CORE
RAM 192 BYTES
OC1
PULSE ACCUMULATOR
PAI
OC2
OC3
OC4
OC5/IC4
IC1
IC2
IC3
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
SERIAL PERIPHERAL INTERFACE (SPI)
SERIAL COMMUNICATION INTERFACE (SCI)
VDD
VSS
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SS
SCK
MOSI
MISO
TxD
RxD
CONTROL
PORT A
PORT B
PORT C
CONTROL
PORT D
PA7
PA0
PB7
PB0
PC7
PC0
STRB/R/W
STRA/AS
PD5
PD0
NOT BONDED ON 40 PIN VERSION

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11E1

Product Preview

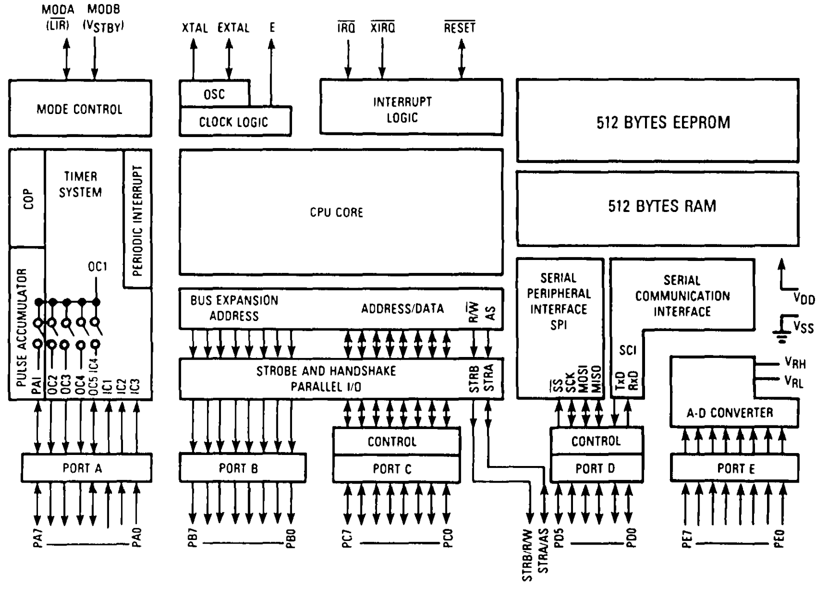
8-Bit Microcontroller Unit

The MC68HC11E1 high-density CMOS (HCMOS) Microcontroller Unit (MCU) contains highly sophisticated, on-chip peripheral capabilities. This high-speed, low-power MCU has a nominal bus speed of 2 MHz and is identical to the MC68HC11E9, except that the chip has no ROM.

Some features of the MC68HC11E1 are as follows:

- Enhanced 16-Bit Timer System with Four-Stage, Programmable Prescaler
- Power-Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 512 Bytes of EEPROM
- 512 Bytes of Static RAM
- Eight-Channel, 8-Bit Analog/Digital Converter
- Available in 52-Pin Plastic Quad (PLCC)

BLOCK DIAGRAM


MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
512 BYTES EEPROM
512 BYTES RAM
COP
TIMER SYSTEM
PERIODIC INTERRUPT
PULSE ACCUMULATOR
OC1
PAI
OC2
OC3
OC4
OC5
IC4
IC1
IC2
IC3
CPU CORE
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SERIAL PERIPHERAL INTERFACE SPI
SERIAL COMMUNICATION INTERFACE
SCI
SS
SCK
MOSI
MISO
TxD
RxD
VDD
VSS
VRH
VRL
A-D CONVERTER
PORT A
PORT B
CONTROL
PORT C
CONTROL
PORT D
PORT E
PA7
PA0
PB7
PB0
PC7
PC0
STRB/R/W
STRA/AS
PD5
PD0
PE7
PE0

This document contains information on a product under development. Motorola reserves the right to discontinue this product without notice.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11E9

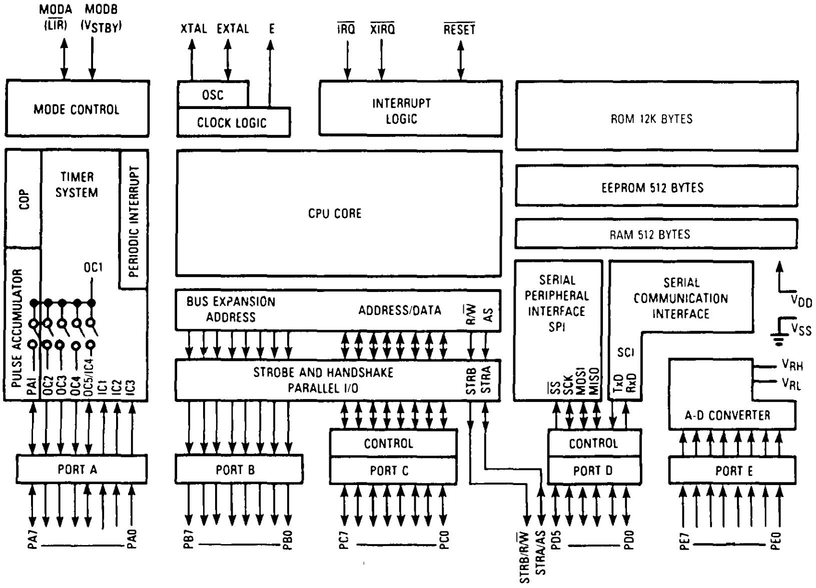
Technical Summary
8-Bit Microcontroller

The MC68HC11E9 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 12K Bytes of ROM
- 512 Bytes of EEPROM
- 512 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

BLOCK DIAGRAM



MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
ROM 12K BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
CPU CORE
EEPROM 512 BYTES
RAM 512 BYTES
PULSE ACCUMULATOR
OC1
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
SERIAL PERIPHERAL INTERFACE SPI
SERIAL COMMUNICATION INTERFACE
VDD
VSS
SCI
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
PAI
OC2
OC3
OC4
OC5/IC4
IC1
IC2
IC3
SS
SCK
MOSI
MISO
TxD
RxD
VRH
VRL
A-D CONVERTER
CONTROL
CONTROL
PORT A
PORT B
PORT C
PORT D
PORT E
PA7
PA0
PB7
PB0
PC7
PC0
STRB/R/W
STRA/AS
PD5
PD0
PE7
PE0

This document contains information on a new product. Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE (MODE0)

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU.

EXPANDED MULTIPLEXED MODE (MODE1)

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{W}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE (MODE0)

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU.

EXPANDED MULTIPLEXED MODE (MODE1)

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{W}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{\text{RESET}}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

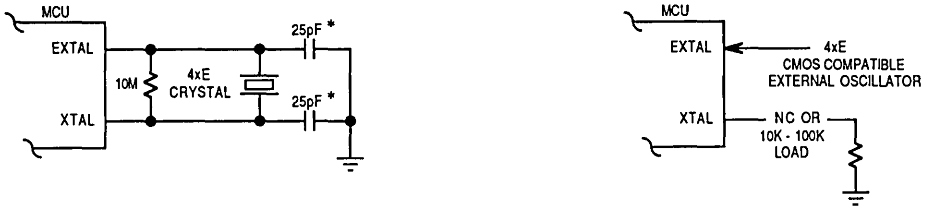
This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins.

$\overline{\text{IRQ}}$

This pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive or level-sensitive triggering is program selectable. This pin is configured to level-sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{\text{IRQ}}$.

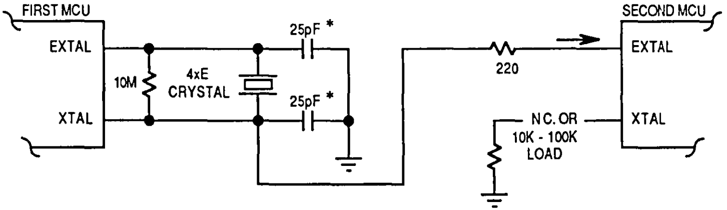
$\overline{\text{XIRQ}}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition



Common Crystal Connections

External Oscillator Connections



*Includes all stray capacitances.

One Crystal Driving Two MCUs

Figure 1. Oscillator Collections

code register is set, and any interrupt is masked until enabled by software. This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{\text{LIR}}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{\text{LIR}}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The mode selections are shown below.

MODB	MODA	MODE SELECTED
1	0	Single Chip
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

R/$\overline{\text{W}}$/STRB

This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides R/$\overline{\text{W}}$ (read-write) function. The R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{\text{W}}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/IC4/and-or OC1	PA3/OC5/IC4/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/$\overline{SS}$	PD5$\overline{SS}$
	STRA	AS
	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE3/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4	PE4/AN4
E-5	PE5/AN5	PE5/AN5
E-6	PE6/AN6	PE6/AN6
E-7	PE7/AN7	PE7/AN7

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for four input capture functions and three output compare functions; four output compare functions and three input capture functions; and a pulse a accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of parallel I/O where the STRA input and STRB output acts as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between the shaded areas (EXT) are for externally addressed memory and I/O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR A AND B

These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.

7	A	0	7	B	0
15			D		0

INDEX REGISTER X (IX)

This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.

15	IX	0

INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.

15	IY	0

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

15	PC	0

STACK POINTER (SP)

The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the accumulators A and B and registers IX and IY can be stored during certain instructions.

15	SP	0

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

7							0
S	X	H	I	N	Z	V	C

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

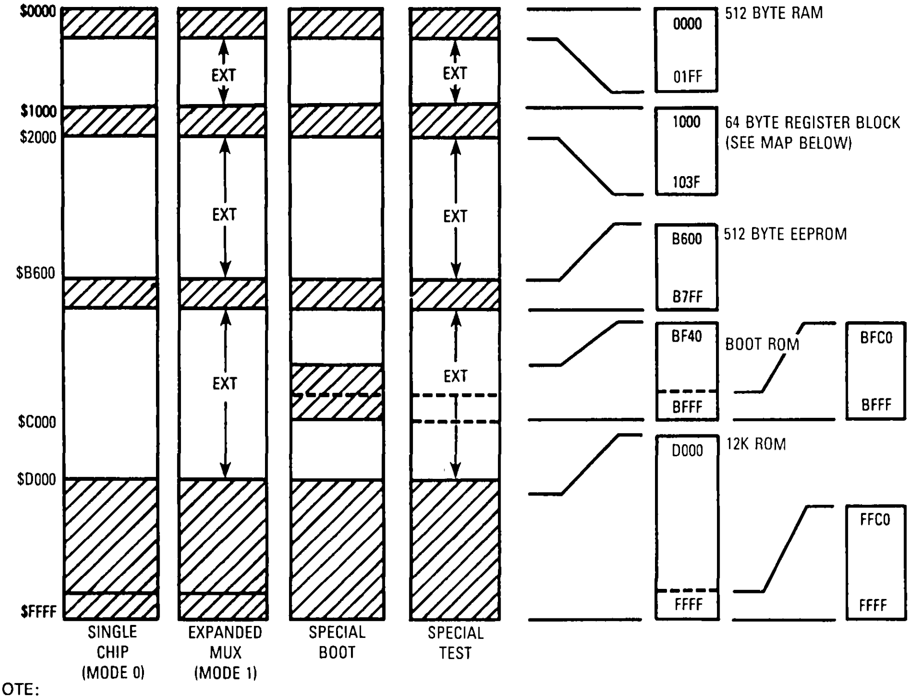
This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or $\overline{\text{XIRQ}}$) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.


$0000
$1000
$2000
$B600
$C000
$D000
$FFFF
EXT
EXT
EXT
EXT
EXT
EXT
SINGLE CHIP (MODE 0)
EXPANDED MUX (MODE 1)
SPECIAL BOOT
SPECIAL TEST
0000
01FF
512 BYTE RAM
1000
103F
64 BYTE REGISTER BLOCK (SEE MAP BELOW)
B600
B7FF
512 BYTE EEPROM
BF40
BFFF
BOOT ROM
BFC0
BFFF
D000
FFFF
12K ROM
FFC0
FFFF

NOTE:

1. Either or both the internal RAM and registers can be remapped to any 4K boundary by software.
2. The EEPROM can be disabled using a control register (CONFIG), which is implemented with EEPROM cells

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TI405	Output Compare 5 Register/
$101F	Bit 7	—	—	—	—	—	—	Bit 0		Output Compare 5 Register/ Input Capture 4 Register
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	I405I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Reg. 1
$1023	OC1F	OC2F	OC3F	OC4F	I405F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Reg. 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Reg. 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Reg 2
$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/05	RTR1	RTR0	PACTL	Pulse Accum. Control Reg
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accum Count Reg
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control

Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035				PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT	EEPROM Block Protect Reg
$1036 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Cir
$103B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG	EEPROM Prog Control Reg
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Reg
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F					NOSEC	NOCOP	ROMON	EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

RESETS

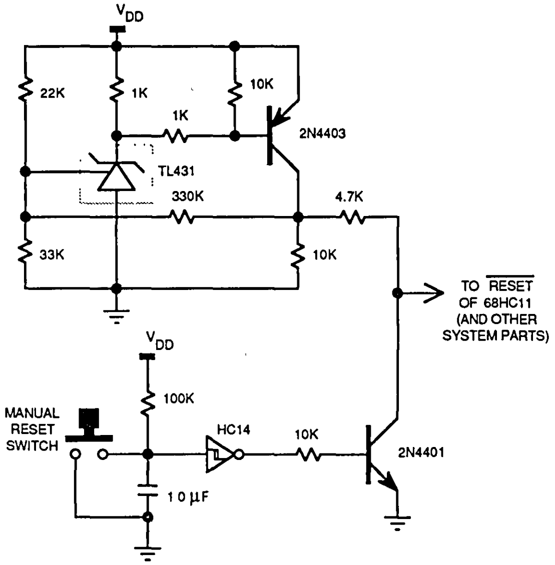
The MCU can be reset four ways: 1) an active low input to the RESET pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

RESET PIN

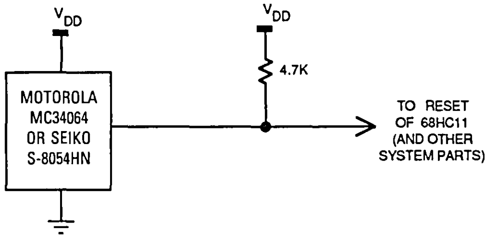
To request an external reset, the RESET pin must be held low for eight E_{cyc} (two E_{cyc} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external RESET pin is low at the end of the power-on delay time, the processor remains in the reset condition until RESET goes high.


V_{DD}
22K
1K
10K
1K
2N4403
TL431
330K
4.7K
33K
10K
TO $\overline{\text{RESET}}$ OF 68HC11 (AND OTHER SYSTEM PARTS)
V_{DD}
100K
MANUAL RESET SWITCH
HC14
10K
2N4401
1 0 µF

Reset Circuit with LVI and RC Delay


V_{DD}
V_{DD}
4.7K
MOTOROLA MC34064 OR SEIKO S-8054HN
TO RESET OF 68HC11 (AND OTHER SYSTEM PARTS)

Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system.

The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

CLOCK MONITOR RESET

The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the $\overline{\text{RESET}}$ pin is driven low to reset the external system.

The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.

Table 2. COP Timeout Periods

CR1	CR0	$E/2^{15}$ Divided By	XTAL = 2^{23} Timeout −1/+15.6 ms	XTAL = 8.0 MHz Timeout −0/+16.4 ms	XTAL = 4.9152 MHz Timeout −0/+26.7 ms	XTAL = 4.0 MHz Timeout −0/+32.8 ms	XTAL = 3.6864 MHz Timeout −0/+35.6 ms
0	0	1	15.625 ms	16.384 ms	26 667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921 6 kHz

INTERRUPTS

There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the $\overline{\text{XIRQ}}$ pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the $\overline{\text{XIRQ}}$ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
*	*		
*	*		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Input Capture 4/Output Compare 5	I Bit	I4O5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	$\overline{\text{XIRQ}}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

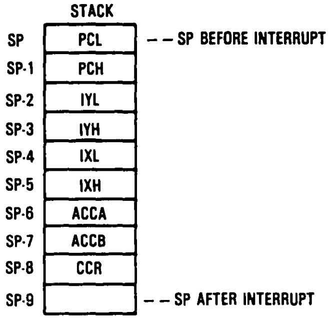


Figure 4. Stacking Order

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be E/2^{13}, E/2^{14}, E/2^{15}, or E/2^{16}.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ or $\overline{\text{RESET}}$. An external interrupt used at $\overline{\text{IRQ}}$ is only efective if the I bit in the CCR is clear. An external interrupt applied at the $\overline{\text{XIRQ}}$ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the $\overline{\text{XIRQ}}$ request. If the X bit is set, the processing will always continue with the instruction immediately following the STOP instruction. A low input to the $\overline{\text{RESET}}$ pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

A restart delay is required if the internal oscillator is being used, to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the $\overline{\text{RESET}}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

WAIT

The wait (WAI) instruction places the MCU in a low-power consumption mode, but the wait mode consumes slightly more power than the stop mode. In the wait mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or $\overline{\text{RESET}}$. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the wait mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces wait mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from $FFFF to $0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has four input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are four 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA3 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Port A pin 3 serves multiple functions. After reset, data direction bit 3 (DDRA3), in the PACTL register is cleared to zero configuring port A pin 3 as an input. Port A pin 3 can then be used as a input capture 4 (IC4), by setting I4/O5 to "one" in the PACTL register. The I4/O5 bit is configured to OC5 (cleared to zero) on reset. If DDRA3 is configured as an output and IC4 is enabled, writes to port A bit 3 causes edges on the PA3 to result in input captures. When the TI405 register is acting as the IC4 capture register it cannot be written to. When PA3 is being used as IC4, writes to TI405 register have no meaning.

TIMER CONTROL REGISTER 2 (TCTL2) $1021

	7	6	5	4	3	2	1	0
	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET	0	0	0	0	0	0	0	0

EDGxB and EDGxA — Input Capture x Edge Control

These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to $FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one. Upon reset, I4/O5 is configured as OC5. The OC5 function overrides DDRA3 to force the Port A pin 3 to be an output whenever OM5:OL5 bits are not 0:0. In all other aspects, OC5 works the same as the other output compares.

TIMER COMPARE FORCE REGISTER (CFORC) $100B

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

	7	6	5	4	3	2	1	0
	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET	0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action

1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set

0 = Has no meaning

Bits 2-0 — Not Implemented

These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M) $100C

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D) $100D

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1) $1020

	7	6	5	4	3	2	1	0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET	0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode

OL2-OL5 — Output Level

These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TIMER INTERRUPT MASK REGISTER 1 (TMSK1)

	7	6	5	4	3	2	1	0
	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I
RESET	0	0	0	0	0	0	0	0

OCxI — Output Compare x Interrupt
1 = Interrupt sequence requested if OCxF = 1 in TFLG1
0 = Interrupt inhibited

ICxI — Input Capture x Interrupt
1 = Interrupt sequence requested if ICxF = 1 in TFLG1
0 = Interrupt inhibited

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5I bit behaves as the input capture 4 interrupt bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 interrupt control bit.

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position.

	7	6	5	4	3	2	1	0
	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F
RESET	0	0	0	0	0	0	0	0

OCxF — Output Compare x Flag
Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected

ICxF — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5F bit behaves as the input capture 4 flag bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 flag.

TIMER INTERRUPT MASK REGISTER 2 (TMSK2) $1024

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG1. Two timer prescaler bits are also included in this register.

	7	6	5	4	3	2	1	0
	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled

RTII — RTI Interrupt Enable
1 = Interrupt requested when RTIF = 1
0 = RTIF interrupt disabled

PAOVI — Pulse Accumulator Overflow Interrupt Enable
1 = Interrupt requested when PAOVF = 1
0 = PAOVF disabled

PAII — Pulse Accumulator Input Interrupt Enable
1 = Interrupt requested when PAIF = 1
0 = PAIF disabled

Bits 3-2 — Not Implemented
These bits always read zero.

PR1 and PR0 — Timer Prescaler Selects
Can only be written to during initialization. Writes are disabled after the first write or after 64 E cycles out of reset.

PR1	PR0	Divide-by-Factor
0	0	1
0	1	4
1	0	8
1	1	16

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2) $1025

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

	7	6	5	4	3	2	1	0
	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

TOF — Timer Overflow
Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag
Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag
Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag
Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented
These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0
	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
- 1 = Output
- 0 = Input only

PAEN — Pulse-Accumulator System Enable
- 1 = Pulse accumulator on
- 0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode
- 1 = Gated time accumulator
- 0 = External even counting

PEDGE — Pulse Accumulator Edge Control

This bit provides clock action along with PAMOD.
- 1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.
- 0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

DDRA3 — Data Directional for Port A Bit 3
- 1 = Output
- 0 = Input only

I4/O5 — Input 4/Output 5
- 1 = Input capture 4 function enabled (No OC5)
- 0 = Output compare 5 function enabled (No IC4)

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

EEPROM PROGRAMMING

The 512 bytes of EEPROM are located at $B600 through $B7FF and have the same read cycle time as the internal ROM. Programming of the EEPROM is controlled by the EEPROM programming control register (PPROG). The EEPROM is disabled when the EEON bit in the system configuration register (CONFIG) is zero. Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz, the efficiency of this charge pump decreases, which increases the time required to program or erase a location. Recommended program and erase time is 10 milliseconds when the E clock is between 2 MHz and should be increased to as much as 20 milliseconds when E clock is between 1 MHz and 2 MHz. When E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. The following paragraphs describe how to program or erase the EEPROM using the PPROG control register.

EEPROM BLOCK PROTECT REGISTER (BPROT) $1035

This register protects inadvertent writes to the CONFIG register and to the 512 bytes of EEPROM. The bits in this register may only be written to "zero" during the first 64 E clock cycles after reset in the normal mode. Once the bits are set to zero, the associated EEPROM section and/or the CONFIG register may be programmed or erased in the normal manner. The EEPROM is only visible if the EEON bit in the CONFIG register is set to "one". The CONFIG register bits may be written to "one" at anytime in the normal mode. In the test or bootstrap mode, the bits in the BPROT register may be set or cleared at anytime. The bits in the BPROT register may be written back to "one" (anytime after the first 64 E clock cycles) to protect the EEPROM and/or the CONFIG register, but can only be cleared in the test or bootstrap modes.

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

	7	6	5	4	3	2	1	0
	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET	0	0	0	1	1	1	1	1

Bits 7–5 — Not Implemented
These bits always read as zero

PTCON — Protect CONFIG Register
1 = Programming/erasure of the CONFIG register disabled
0 = Programming/erasure of the CONFIG register allowed

BPRT3–BPRT0 — Block Protect
When set, these bits protect a block of EEPROM from programming and erasure; when cleared, these bits allow programming and erase of the associated block.

Bit	Block Protected	Size
BPRT0	$B600-$B61F	32 Bytes
BPRT1	$B620-$B65F	64 Bytes
BPRT2	$B660-$B6DF	128 Bytes
BPRT3	$B6E0-$B7FF	288 Bytes

ERASING THE EEPROM

Erasure of the EEPROM is controled by bit settings in PPROG, and the appropriate bits in the BPROT register must also be cleared before the EEPROM can be changed. Programs can be written to perform bulk, row, or byte erase In bulk erase, all 512 bytes of the EEPROM are erased. In row erase, 16 bytes ($B600-$B60F, $B610-$B61F), etc) are erased. Other MCU operations can continue to be performed during erasing provided the operations do not include reads of data from EEPROM.

PROGRAMMING EEPROM

During programming, the ROW and BYTE bits are not used. If the E clock frequency is 1 MHz or less, the CSEL bit in the OPTION register must be set. Zeros must be erased by a separate erase operation before programming. Other MCU operations can continue to be performed during programming provided the operations do not include reads of data from EEPROM.

EEPROM PROGRAMMING CONTROL REGISTER (PPROG) $103B

This 8-bit register is used to control programming and erasure of the EEPROM This register is cleared on reset so the EEPROM is configured for normal reads.

	7	6	5	4	3	2	1	0
	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET	0	0	0	0	0	0	0	0

ODD — Program Odd Rows (TEST)

EVEN — Program Even Rows (TEST)

Bit 5 — Not Implemented
This bit always reads zero.

BYTE — Byte Erase Select
This bit overrides the ROW bit.
1 = Erase only one byte
0 = Row or bulk erase

ROW — Row Erase Select
If BYTE bit = 1, ROW has no meaning.
1 = Row erase
0 = Bulk or byte erase

ERASE — Erase Mode Select
1 = Erase mode
0 = Normal read or program

EELAT — EEPROM Latch Control
1 = EEPROM Address and data configured for programmming/erasing
0 = EEPROM Address and data configured for read mode

EEPGM — EEPROM Programming Voltage Enable
1 = Programming voltage turned on
0 = Programming voltage turned off

NOTE

A strict register access sequence must be followed to allow successful programming and erase operations. The following procedures for modifying the EEPROM and CONFIG register detail the sequence. If an attempt is made to set both the EELAT and EEPGM bits in the same write cycle and if this attempt occurs before the required write cycle with the EELAT bit set, then neither is set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation in progress is not disturbed. If no EEPROM address is written between when EELAT is set and EEPGM is set, then no program or erase operation takes place. These safeguards were included to prevent accidental EEPROM changes in cases of program runaway.

ERASING THE CONFIG REGISTER

Erasing the CONFIG register follows the same procedures as that used for the EEPROM including bulk, byte, and row erase. The CONFIG register may be programmed or erased while the MCU is operating in any mode depending on the setting of bit 4 in BPROT.

PROGRAMMING THE CONFIG REGISTER

Programming the CONFIG register follows the same procedures as that used for the EEPROM except the CONFIG register address is used.

SYSTEM CONFIGURATION REGISTER (CONFIG) $103F

The CONFIG is implemented in EEPROM cells and controls the presence of ROM and EEPROM in the memory map and enables the COP watchdog system.

7	6	5	4	3	2	1	0
0	0	0	0	NOSEC	NOCOP	ROMON	EEON

Bits 7-4 - Not Implemented
These bits are always read as zero.

NOSEC — Security Mode Disable Bit
This bit is only implemented if it is specifically requested with the submission of mask ROM information.
1 = Disable security mode
0 = Enable security mode

NOCOP — COP System Disable
1 = COP watchdog system disabled
0 = COP watchdog system enabled

ROMON — Enable On-Chip ROM
When this bit is programmed to "zero", the 12K ROM is disabled, and that memory space becomes externally accessed space. In the single-chip mode, the internal 12K ROM is enabled regardless of the state of the ROMON bit.

EEON — Enable On-Chip EEPROM
When this bit is programmed to "zero", the 512-byte EEPROM is disabled, and that memory space becomes externally accessed space.

SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins: PD0 for receive data (RxD) and PD1 for the transmit data (TxD). The baud-rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/reception of a message;
2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR) $102F

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1) $102C

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

	7	6	5	4	3	2	1	0
	R8	T8	0	M	WAKE	0	0	0
RESET	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.

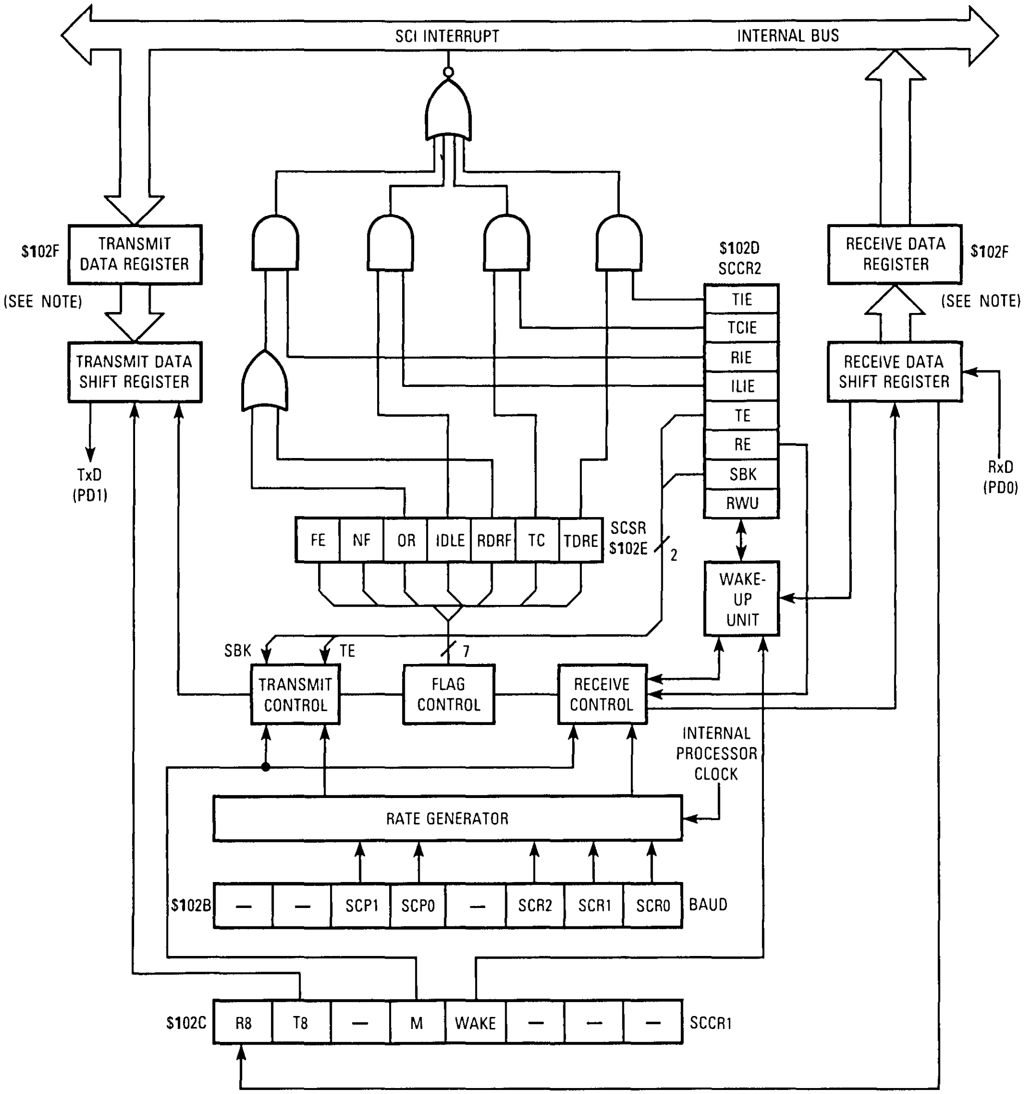
T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.

Bit 5 — Not Implemented
This bit always reads zero.

M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line

Bits 2-0 — Not Implemented
These bits always read zero.


SCI INTERRUPT
INTERNAL BUS
$102F
TRANSMIT DATA REGISTER
(SEE NOTE)
TRANSMIT DATA SHIFT REGISTER
TxD (PD1)
$102D SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
RECEIVE DATA REGISTER
$102F
(SEE NOTE)
RECEIVE DATA SHIFT REGISTER
RxD (PD0)
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR $102E
2
WAKE-UP UNIT
SBK
TE
7
TRANSMIT CONTROL
FLAG CONTROL
RECEIVE CONTROL
INTERNAL PROCESSOR CLOCK
RATE GENERATOR
$102B
—
—
SCP1
SCP0
—
SCR2
SCR1
SCR0
BAUD
$102C
R8
T8
—
M
WAKE
—
—
—
SCCR1

NOTE. The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read

Figure 5. SCI Block Diagram

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
- 1 = SCI interrupt if TDRE = 1
- 0 = TDR interrupts disabled

TCIE — Transmit-Complete Interrupt Enable
- 1 = SCI interrupt if TC = 1
- 0 = TC interrupts disabled

RIE — Receive Interrupt Enable
- 1 = SCI interrupt if RDRF or OR = 1
- 0 = RDRF or OR interrupt disabled

ILIE — Idle-Line Interrupt Enable
- 1 = SCI interrupt if IDLE = 1
- 0 = IDLE interrupts disabled

TE — Transmit Enable
- 1 = Transmit shift register output is applied to the TxD line
- 0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
- 1 = Receiver enabled
- 0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited

RWU — Receiver Wake Up

When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the WAKE bit is zero, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If this bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

	7	6	5	4	3	2	1	0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET	1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty
- 1 = Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register
- 0 = Cleared by a read of SCSR (with TDRE = 1) followed by a write to SCDR

TC — Transmit Complete
- 1 = Automatically set when all data frame, preamble, or break condition transmissions are complete
- 0 = Cleared by a read of SCSR (with TC = 1) followed by a write to SCDR

RDRF — Receive Data Register Full
- 1 = Automatically set when a character is transferred from the receiver shift register to the SCDR
- 0 = Cleared by a read of SCSR (with RDRF = 1) followed by a read of SCDR

IDLE — Idle-Line Detect

This bit is inhibited while RWU = 1.
- 1 = Automatically set when the receiver serial input becomes idle after having been active
- 0 = Cleared by a read of SCSR (with IDLE = 1) followed by a read of SCDR

OR — Overrun Error
- 1 = Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read
- 0 = Cleared by a read of SCSR (with OR = 1) followed by a read of SCDR

NF — Noise Flag
- 1 = Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame
- 0 = Cleared by a read of SCSR (with NF = 1) followed by a write to SCDR

FE — Framing Error
- 1 = Automatically set when a logic 0 is detected where a stop bit was expected
- 0 = Cleared by a read of SCSR (with FE = 1) followed by a read of SCDR

Bit 0 — Not Implemented

This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

	7	6	5	4	3	2	1	0
	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET	0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)

This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.

Bit 6 — Not Implemented

This bit always reads zero.

SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects

These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)

This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

SCR2-SCR0 — SCI Baud-Rate Selects

These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt if SPIF = 1
0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

DWOM — Port D Wire-OR Mode Option
This bit affects all six port D pins together.
1 = Port D outputs act as open-drain outputs
0 = Port D outputs are normal CMOS outputs

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	1	0		131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

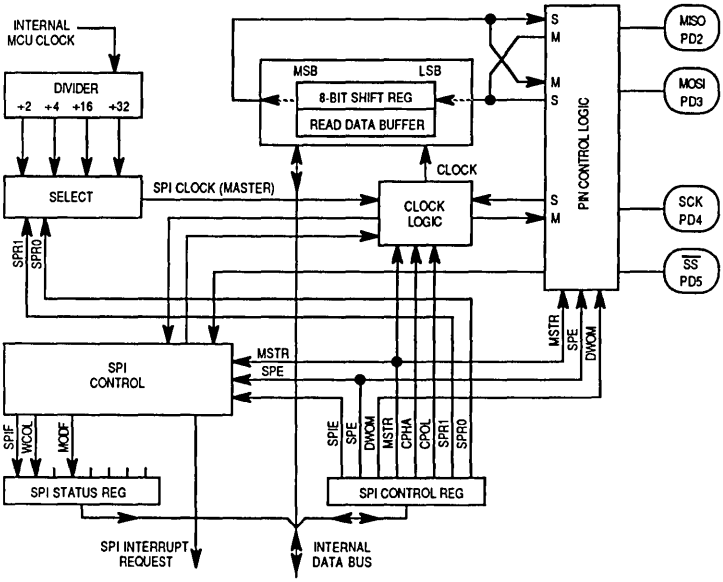

INTERNAL MCU CLOCK
DIVIDER
÷2 ÷4 ÷16 ÷32
SELECT
SPR1
SPR0
SPI CLOCK (MASTER)
MSB
LSB
8-BIT SHIFT REG
READ DATA BUFFER
CLOCK
CLOCK LOGIC
S
M
M
S
S
M
PIN CONTROL LOGIC
MISO PD2
MOSI PD3
SCK PD4
SS PD5
MSTR
SPE
DWOM
SPI CONTROL
MSTR
SPE
SPIF
WCOL
MODF
SPIE
SPE
DWOM
MSTR
CPHA
CPOL
SPR1
SPR0
SPI STATUS REG
SPI CONTROL REG
SPI INTERRUPT REQUEST
INTERNAL DATA BUS

Figure 6. SPI Block Diagram

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
This bit selects the polarity of the SCK clock.
1 = SCK line idles high
0 = SCK line idles low

CPHA — Clock Phase
This bit selects one of two fundamentally different clock protocols. Refer to Figure 7.
If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{SS}$ is low and ends when the SPIF flag gets set.

SPR1 and SPR0 — SPI Clock Rate Select
These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR) $1029

	7	6	5	4	3	2	1	0
	SPIF	WCOL	0	MODF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag
1 = Automatically set when data transfer is complete between processor and external device
0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision
1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred
0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented
This bit always reads zero.

MODF — Mode Fault
This bit indicates the possibility of a multi-master conflict for system control and therefore allows a proper exit from system operation to a reset or default system state.
1 = Automatically set when a master device has its $\overline{SS}$ pin pulled low
0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented
These bits always read zero.

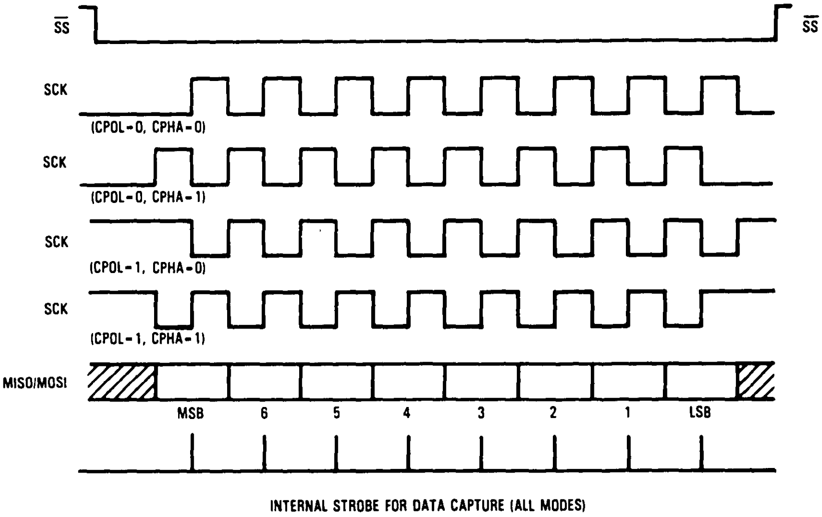


Figure 7. Data Clock Timing Diagram

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.
2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16×16	FDIV
Integer Divide 16×16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8×8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if = zero	BEQ
Branch if ≥ zero	BGE
Branch if) zero	BGT
Branch if Higher	BHI
Branch if ≤ Zero	BLE
Branch if Lower or Same	BLS
Branch if < Zero	BLT
Branch if Minus	BMI
Branch if not = Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

— Continued—

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

OPCODE MAP SUMMARY

Table 6 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

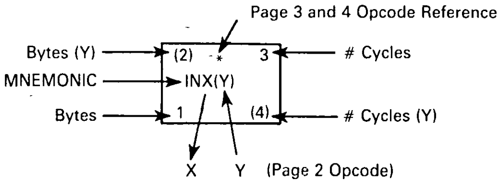
Table 6. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
HI / LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LOW
0 0000	TEST 1	2 SBA 1	3 BRA 2	(2) 3 TSX(Y) 1 (4)	2 NEGA 1	2 NEGB 1	(3) 6 NEG 2 (7)	6 NEG 3	2 SUBA 2	3 SUBA 2	(3) 4 SUBA 2 (5)	4 SUBA 3	2 SUBB 2	3 SUBB 2	(3) 4 SUBB 2 (5)	4 SUBB 3	0
1 0001	2 NOP 1	2 CBA 1	3 BRN 2	3 INS 1					2 CMPA 2	3 CMPA 2	(3) 4 CMPA 2 (5)	4 CMPA 3	2 CMPB 2	3 CMPB 2	(3) 4 CMPB 2 (5)	4 CMPB 3	1
2 0010	41 IDIV 1	6 BRSET 4	3 BHI 2	4 PULA 1					2 SBCA 2	3 SBCA 2	(3) 4 SBCA 2 (5)	4 SBCA 3	2 SBCB 2	3 SBCB 2	(3) 4 SBCB 2 (5)	4 SBCB 3	2
3 0011	41 FDIV 1	6 BRCLR 4	3 BLS 2	4 PULB 1	2 COMA 1	2 COMB 1	(3) 6 COM 2 (7)	6 COM 3	4 *SUBD 3	5 *SUBD 2	(3) 6 *SUBD 2 (7)	6 *SUBD 3	4 ADDD 3	5 ADDD 2	(3) 6 ADDD 2 (7)	6 ADDD 3	3
4 0100	3 LSRD 1	6 BSET 3	3 (BHS) BCC 2	3 DES 1	2 LSRA 1	2 LSRB 1	(3) 6 LSR 2 (7)	6 LSR 3	2 ANDA 2	3 ANDA 2	(3) 4 ANDA 2 (5)	4 ANDA 3	2 ANDB 2	3 ANDB 2	(3) 4 ANDB 2 (5)	4 ANDB 3	4
5 0101	3 (LSLD) ASLD 1	6 BCLR 3	3 (BLO) BCS 2	(2) 3 TX(Y)S 1 (4)					2 BITA 2	3 BITA 2	(3) 4 BITA 2 (5)	4 BITA 3	2 BITB 2	3 BITB 2	(3) 4 BITB 2 (5)	4 BITB 3	5
6 0110	2 TAP 1	2 TAB 1	3 BNE 2	3 PSHA 1	2 RORA 1	2 RORB 1	(3) 6 ROR 2 (7)	6 ROR 3	2 LDAA 2	3 LDAA 2	(3) 4 LDAA 2 (5)	4 LDAA 3	2 LDBB 2	3 LDBB 2	(3) 4 LDBB 2 (5)	4 LDBB 3	6
7 0111	2 TPA 1	2 TBA 1	3 BEQ 2	3 PSHB 1	2 ASRA 1	2 ASRB 1	(3) 6 ASR 2 (7)	6 ASR 3		3 STAA 2	(3) 4 STAA 2 (5)	4 STAA 3		3 STBB 2	(3) 4 STBB 2 (5)	4 STBB 3	7
8 1000	(2) (3) INX(Y) (1) (4)	PAGE 2	3 BVC 2	(2) 5 PULX(Y) 1 (6)	2 ASLA 1	2 ASLB 1	(3) 6 ASL 2 (7)	6 ASL 3	2 EORA 2	(3) 3 EORA 2 (5)	4 EORA 2	4 EORA 3	2 EORB 2	3 EORB 2	(3) 4 EORB 2 (5)	4 EORB 3	8
9 1001	(2) 3 DEX(Y) 1 (4)	2 DAA 1	7 BVS 2	5 RTS 1	2 ROLA 1	2 ROLB 1	(3) 6 ROL 2 (7)	6 ROL 3	2 ADCA 2	3 ADCA 2	(3) 4 ADCA 2 (5)	4 ADCA 3	2 ADCB 2	3 ADCB 2	(3) 4 ADCB 2 (5)	4 ADCB 3	9
A 1010	2 CLV 1	PAGE 3	3 BPL 2	(2) 3 ABX(Y) 1 (4)	2 DECA 1	2 DECB 1	(3) 6 DEC 2 (7)	6 DEC 3	2 DRAA 2	3 DRAA 2	(3) 4 ORAA 2 (5)	4 ORAA 3	2 ORAB 2	3 ORAB 2	(3) 4 ORAB 2 (5)	4 ORAB 3	A
B 1011	2 SEV 1	2 ABA 1	3 BMI 2	12 RTI 1					2 ADDA 2	3 ADDA 2	(3) 4 ADDA 2 (5)	4 ADDA 3	2 ADDB 2	3 ADDB 2	(3) 4 ADDB 2 (5)	4 ADDB 3	B
C 1100	2 CLC 1	(4) 7 BSET 3 (8)	3 BGE 2	(2) 4 PSHX(Y) 1 (5)	2 INCA 1	2 INCB 1	(3) 6 INC 2 (7)	6 INC 3	4 CPX(Y) 3	5 CPX(Y) 2	(3) 6 *CPX(Y) 2 (7)	6 CPX(Y) 3	3 LDD 3	4 LDD 2	(3) 5 LDD 2 (6)	5 LDD 3	C
D 1101	2 SEC 1	(4) 7 BCLR 3 (8)	3 BLT 2	10 MUL 1	2 TSTA 1	2 TSTB 1	(3) 6 TST 2 (7)	6 TST 3	6 BSR 2	5 JSR 2	(3) 6 JSR 2 (7)	6 JSR 3	PAGE 4	4 STD 2	(3) 5 STD 2 (6)	5 STD 3	D
E 1110	2 CLI 1	(5) 7 BRSET 4 (8)	3 BGT 2	12 WAI 1			(3) 3 JMP 2 (4)	3 JMP 2	3 LDS 3	4 LDS 2	(3) 5 LDS 2 (6)	5 LDS 3	3 LDX(Y) 3	4 LDX(Y) 2	(3) 5 LDX(Y) 2 (6)	5 *LDX(Y) 3	E
F 1111	2 SEI 1	(5) 7 BRCLR 4 (8)	3 BLE 2	14 SWI 1	2 CLRA 1	2 CLRB 1	(3) 6 CLR 2 (7)	6 CLR 3	(2) 3 XGDX(Y) 1 (4)	4 STS 2	(3) 5 STS 2 (6)	5 STS 3	2 STOP 1	(3) 4 STX(Y) 2 (5)	(3) 5 STX(Y) 2 (6)	(4) 5 *STX(Y) 3 (6)	F

*Page 3 and 4 Opcode Reference

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC11E9 MC68HC11E9V MC68HC11E9M	T_A	T_L to T_H −40 to 85 −40 to 105 −40 to 125	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC)	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

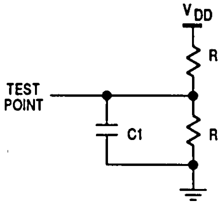
where K is a constant pertaining to the particular part K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

Characteristic		Symbol	Min	Max	Unit
Output Voltage	All Outputs	V_{OL}	—	0.1	V
I_{Load} = ± 10.0 µA (see Note 1)	All Outputs Except $\overline{\text{RESET}}$ and MODA	V_{OH}	$V_{DD}-0.1$	—	
Output High Voltage I_{Load} = − 0.8 mA, V_{DD} = 4.5 V (see Note 1)	All Outputs Except $\overline{\text{RESET}}$, XTAL, and MODA	V_{OH}	$V_{DD}-0.8$	—	V
Output Low Voltage I_{Load} = 1.6 mA	All Outputs Except XTAL	V_{OL}	—	0.4	V
Input High Voltage	All Inputs Except $\overline{\text{RESET}}$	V_{IH}	$0.7 \times V_{DD}$	V_{DD}	V
	$\overline{\text{RESET}}$		$0.8 \times V_{DD}$	V_{DD}	
Input Low Voltage	All Inputs	V_{IL}	V_{SS}	$0.2 \times V_{DD}$	V
I/O Ports, Three-State Leakage V_{in} = V_{IH} or V_{IL}	PA3, PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	I_{OZ}	—	± 10	µA
Input Current (see Note 2)		I_{in}			µA
V_{in} = V_{DD} or V_{SS}	PA0-PA3, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$		—	± 1	
V_{in} = V_{DD} or V_{SS}	MODB/V_{STBY}		—	± 10	
RAM Standby Voltage	Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	µA
Total Supply Current (see Note 3)					
RUN:		I_{DD}			
Single Chip			—	15	mA
Expanded Multiplexed			—	27	mA
WAIT:		W_{IDD}			
All Peripheral Functions Shut Down					
Single-Chip Mode			—	6	mA
Expanded Multiplexed Mode			—	10	mA
STOP:		S_{IDD}			
No Clocks, Single-Chip Mode			—	100	µA
Input Capacitance	PA0-PA3, PE0-PE7, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL	C_{in}	—	8	pF
	PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$		—	12	
Power Dissipation	Single-Chip Mode	P_D	—	85	mW
	Expanded-Multiplexed Mode		—	150	

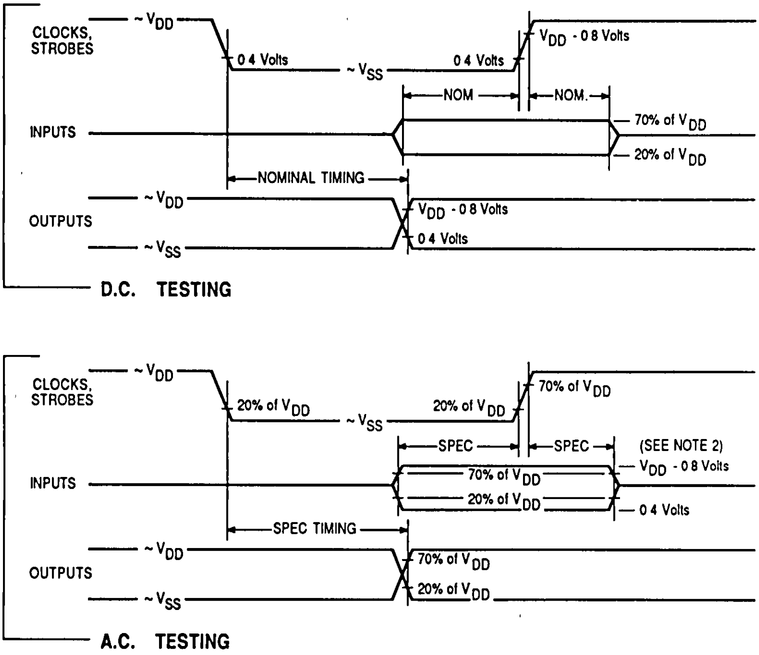
NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL} \leq 0.2$ V,
   $V_{IH} \geq V_{DD} - 0.2$ V,
   No dc loads,
   EXTAL is driven with a square wave, and
   t_{cyc} = 476.5 ns.


VDD
R2
TEST POINT
C1
R1

Equivalent Test Load[1]

Pins	R1	R2	C1
PA3-PA7 PB0-PB7 PC0-PC7 PD0, PD5 E, AS, R/W	3.26K	2.38K	90pF
PD1-PD4	3 26K	2 38K	200pF


CLOCKS, STROBES
~VDD
0 4 Volts
~VSS
0 4 Volts
VDD - 0 8 Volts
NOM
NOM.
INPUTS
70% of VDD
20% of VDD
NOMINAL TIMING
OUTPUTS
~VDD
~VSS
VDD - 0 8 Volts
0 4 Volts
D.C. TESTING
CLOCKS, STROBES
~VDD
20% of VDD
~VSS
20% of VDD
70% of VDD
SPEC
SPEC
(SEE NOTE 2)
INPUTS
70% of VDD
20% of VDD
VDD - 0 8 Volts
0 4 Volts
SPEC TIMING
OUTPUTS
~VDD
~VSS
70% of VDD
20% of VDD
A.C. TESTING

NOTES:

1. Full test loads are applied during all ac electrical test and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and V_{DD} − 0 8 volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

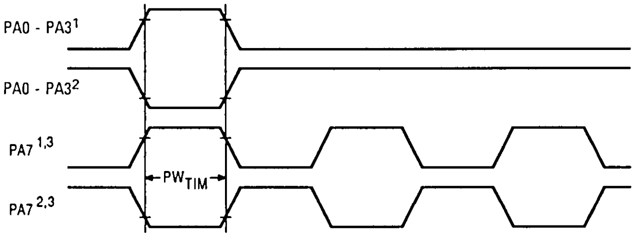
Figure 8. Test Methods

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and 13) $t_{PCS} = 1/4\ t_{cyc} - 50$ ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time; May be Preempted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) $PW_{IRQ} = t_{cyc} + 20$ ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) $PW_{TIM} = t_{cyc} + 20$ ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:

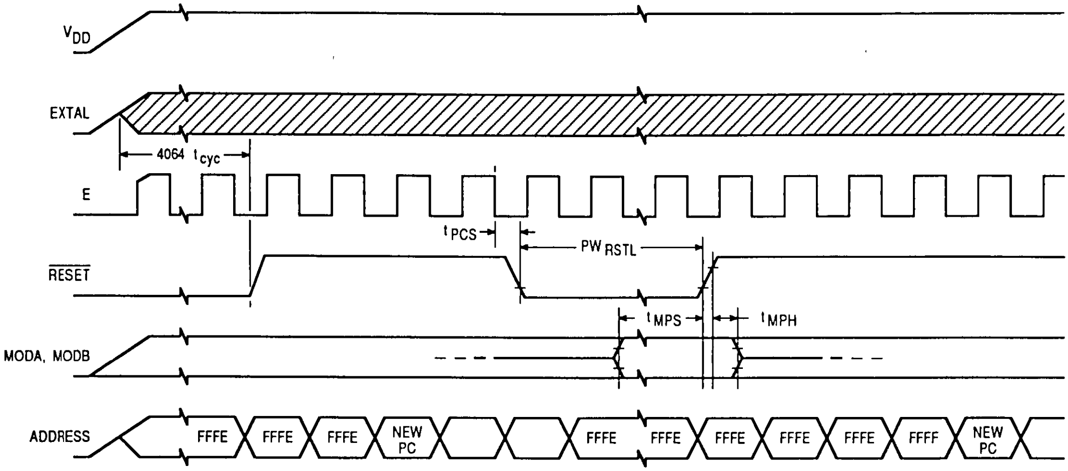
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



NOTES

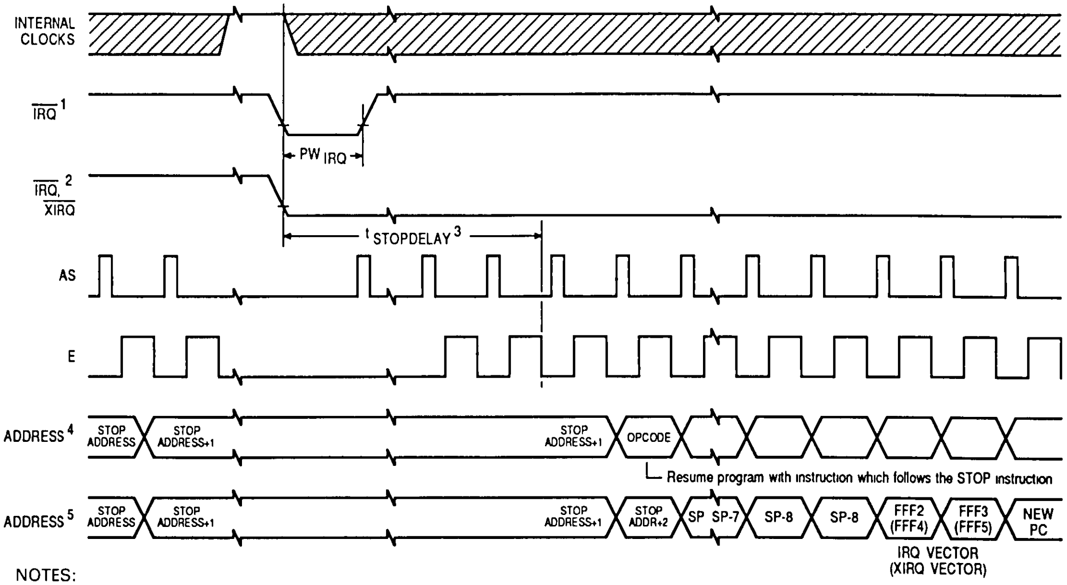
1 Rising edge sensitive input
2 Falling edge sensitive input.
3 Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 9. Timer Inputs Timing Diagram


V_{DD}
EXTAL
4064 t_{cyc}
E
t_{PCS}
PW_{RSTL}
RESET
t_{MPS}
t_{MPH}
MODA, MODB
ADDRESS
FFFE
FFFE
FFFE
NEW PC
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
NEW PC

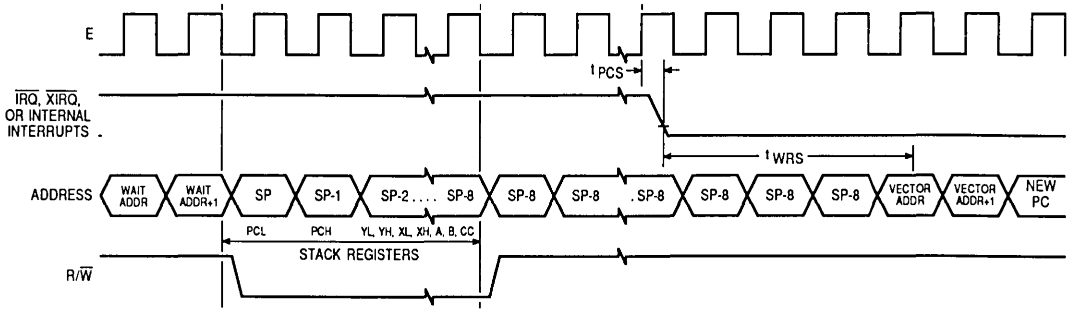
NOTE Refer to Table 9-7 for pin states during RESET

Figure 10. POR External Reset Timing Diagram


INTERNAL CLOCKS
IRQ 1
PW_{IRQ}
IRQ, 2
XIRQ
$t_{STOPDELAY}$ 3
AS
E
ADDRESS 4
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
OPCODE
Resume program with instruction which follows the STOP instruction
ADDRESS 5
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
STOP ADDR+2
SP
SP-7
SP-8
SP-8
FFF2 (FFF4)
FFF3 (FFF5)
NEW PC
IRQ VECTOR
(XIRQ VECTOR)

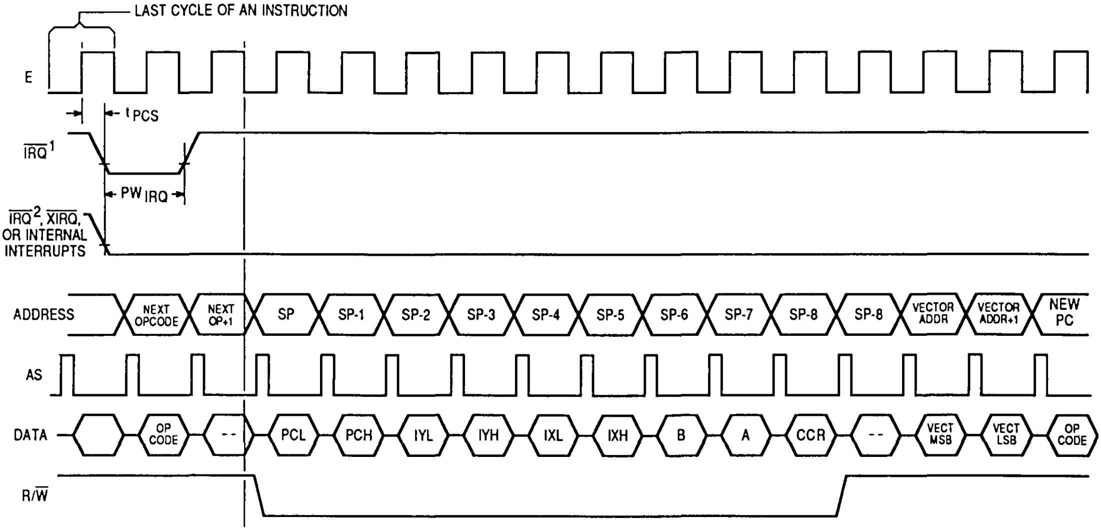
NOTES:
1. Edge sensitive IRQ pin (IRQE bit = 1)
2. Level sensitive IRQ pin (IRQE bit = 0)
3. $t_{\overline{ST}OPDELAY}$ = 4064 t_{cyc} if DLY bit = 1 or 4 t_{cyc} if DLY = 0
4. XIQ with X bit in CCR = 1.
5. IRQ or (XIRQ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram


E
IRQ, XIRQ, OR INTERNAL INTERRUPTS
tPCS
tWRS
ADDRESS
WAIT ADDR
WAIT ADDR+1
SP
SP-1
SP-2.... SP-8
SP-8
SP-8
.SP-8
SP-8
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
PCL
PCH
YL, YH, XL, XH, A, B, CC
STACK REGISTERS
R/W

NOTES
1 Refer to Table 9-7 for pin states during WAIT
2 RESET will also cause recovery from WAIT

Figure 12. WAIT Recovery from Interrupt Timing Diagram


LAST CYCLE OF AN INSTRUCTION
E
tPCS
IRQ1
PWIRQ
IRQ2, XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
NEXT OPCODE
NEXT OP+1
SP
SP-1
SP-2
SP-3
SP-4
SP-5
SP-6
SP-7
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
AS
DATA
OP CODE
--
PCL
PCH
IYL
IYH
IXL
IXH
B
A
CCR
--
VECT MSB
VECT LSB
OP CODE
R/W

NOTES
1 Edge sensitive IRQ pin (IRQE bit = 1)
2 Level sensitive IRQ pin (IRQE bit = 0)

Figure 13. Interrupt Timing Diagram

PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 16, 18, and 19)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D $t_{PWD} = 1/4\ t_{cyc} + 90$ ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 17 and 18)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 17 and 18)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB $t_{DEB} = 1/4\ t_{cyc} + 100$ ns (See Figure 16, 18, 19, and 20)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 18, 19, 20)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 20)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 20)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 20)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:
1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

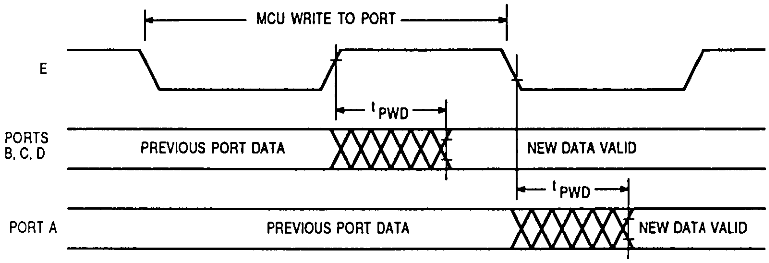
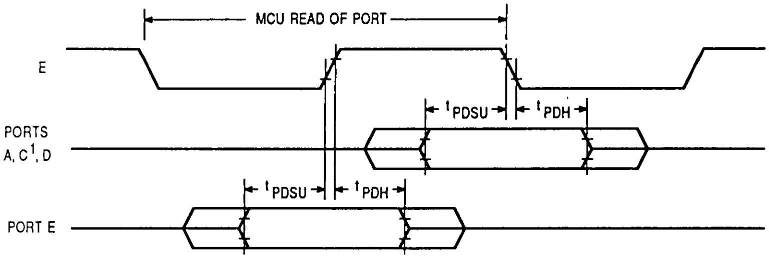

MCU WRITE TO PORT
E
t PWD
PORTS B, C, D
PREVIOUS PORT DATA
NEW DATA VALID
t PWD
PORT A
PREVIOUS PORT DATA
NEW DATA VALID

Figure 14. Port Write Timing Diagram


MCU READ OF PORT
E
t PDSU
t PDH
PORTS A, C1, D
t PDSU
t PDH
PORT E

NOTE1 For non-latched operation of Port C

Figure 15. Port Read Timing Diagram

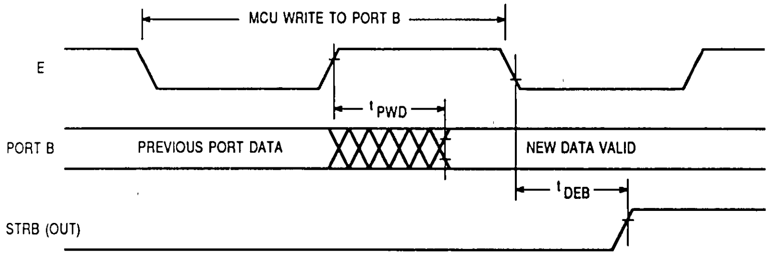

MCU WRITE TO PORT B
E
t PWD
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
t DEB
STRB (OUT)

Figure 16. Simple Output Strobe Timing Diagram

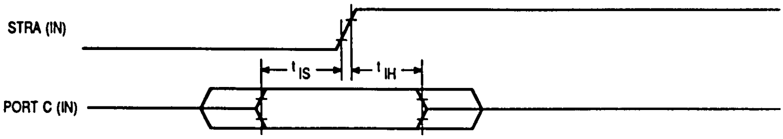
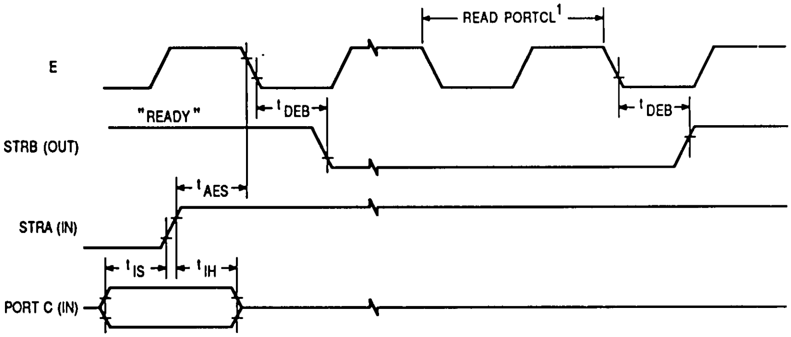
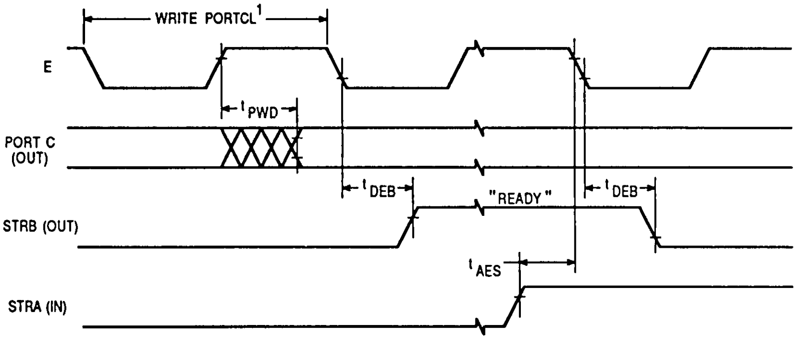

STRA (IN)
PORT C (IN)
tIS
tIH

Figure 17. Simple Input Strobe Timing Diagram


READ PORTCL1
E
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)
tIS
tIH
PORT C (IN)

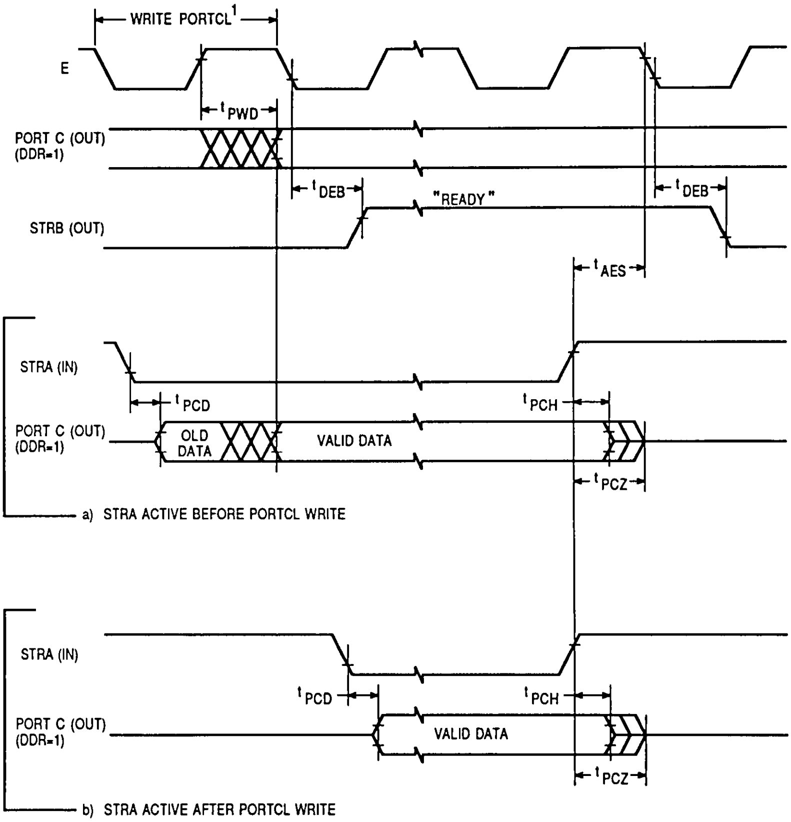
NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 18. Port C Input Handshake Timing Diagram


WRITE PORTCL1
E
tPWD
PORT C (OUT)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)

NOTES.
1 After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 19. Port C Output Handshake Timing Diagram


WRITE PORTCL¹
E
tPWD
PORT C (OUT)
(DDR=1)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)
tPCD
tPCH
PORT C (OUT)
(DDR=1)
OLD DATA
VALID DATA
tPCZ
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
tPCD
tPCH
PORT C (OUT)
(DDR=1)
VALID DATA
tPCZ
b) STRA ACTIVE AFTER PORTCL WRITE

NOTES
1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	$V_{DD} + 0.1$	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	$V_{SS} - 0.1$	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion: a. E Clock b. Internal RC Oscillator	 — —	 32 —	 — $t_{cyc} + 32$	 t_{cyc} µs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time: a. E Clock b. Internal RC Oscillator	 — —	 12 —	 — 12	 t_{cyc} µs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7 V_{RL}, V_{RH}	— —	— —	400 1.0	nA µA

NOTES:

1. Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns$)$ see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns$)$ see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES.

1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1/8 t_{cyc} in the above formulas where applicable:
   (a) $(1\text{-DC}) \times 1/4\ t_{cyc}$
   (b) $\text{DC} \times 1/4\ t_{cyc}$
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, see Figure 22)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

NOTE:

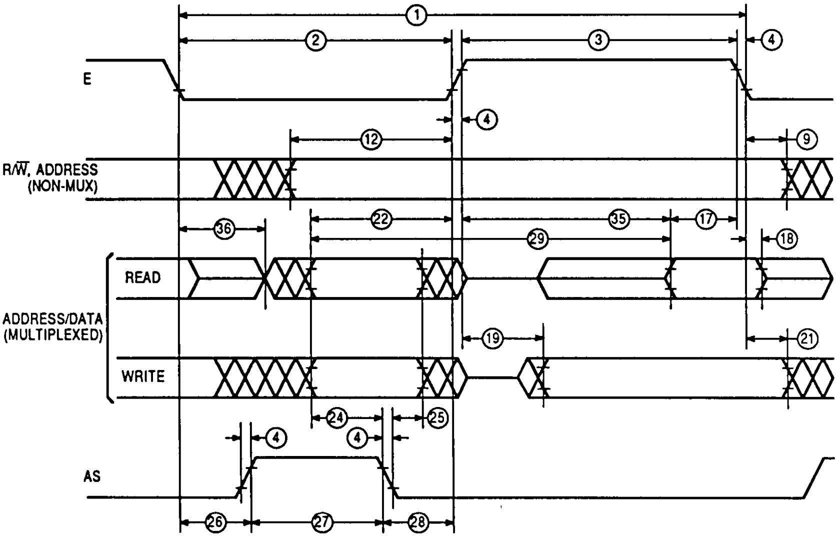
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted

EEPROM CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic		Temperature Range			Unit
		−40 to 85°C	−40 to 105°C	−40 to 125°C	
Programming Time (see Note 1)	Under 1.0 MHz with RC Oscillator Enabled	10	15	20	ms
	1.0 to 2.0 MHz with RC Oscillator Disabled	20	Must Use RC	Must Use RC	
	2.0 MHz (or Anytime RC Oscillator Enabled)	10	15	20	
Erase Time (see Note 1)	Byte, Row, and Bulk	10	10	10	ms
Write/Erase Endurance (see Note 2)		10,000	10,000	10,000	Cycles
Data Retention (see Note 2)		10	10	10	Years

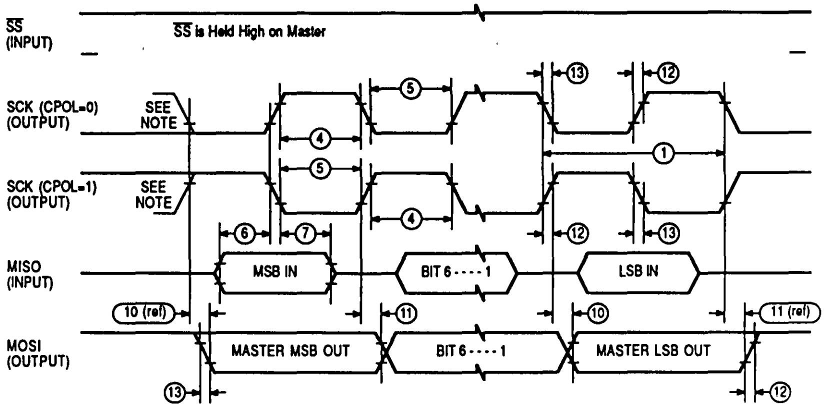
NOTES:

1. The RC oscillator must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. See current quarterly Reliability Monitor report for current failure rate information.



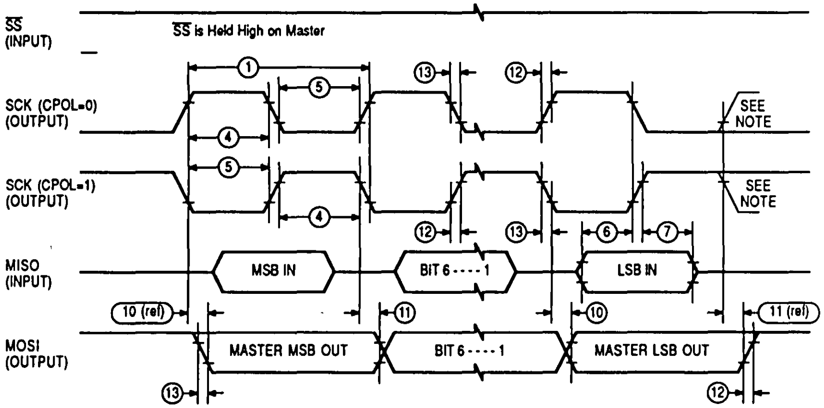
NOTE: Measurement points shown are 20% and 70% V_{DD}

Figure 21. Expansion Bus Timing Diagram


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
10 (ref)
11 (ref)
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

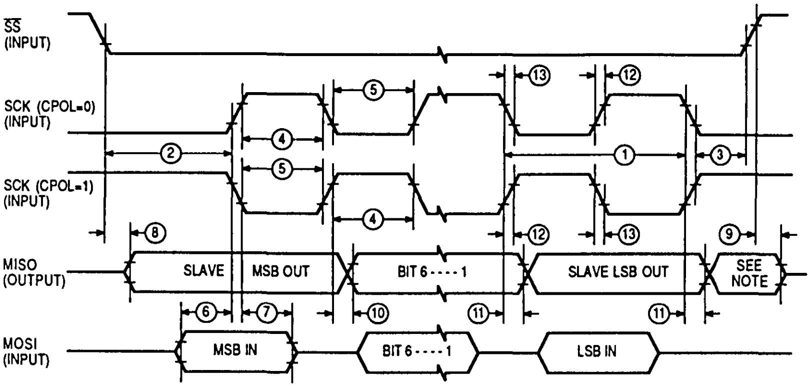
a) SPI MASTER TIMING (CPHA=0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
10 (ref)
11 (ref)
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

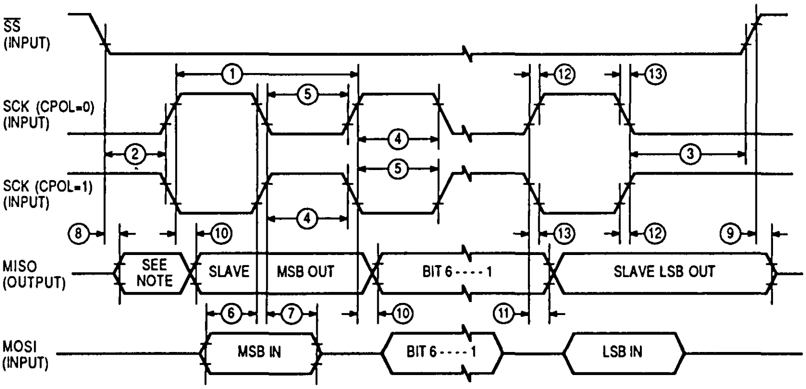
b) SPI MASTER TIMING (CPHA=1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE: Not defined but normally MSB of character just received.

c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

MECHANICAL DATA

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MS-DOS/PC-DOS disk file (360K)
EPROM(s): three 2532/2732 or two 2764

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field-service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MS-DOS™/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

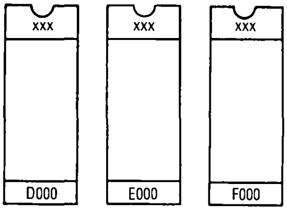
In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5 1/4-inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M68HC11 cross assemblers and linkers on IBM PC-style machines.

EPROMs

Three 2532/2732 or two 2764 type EPROM(s), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure 23 illustrates the markings for the three 2532/2732 EPROMs required to contain the customer's program.

All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Figure 23. EPROM Marking

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program ***customer supplied*** blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum-order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

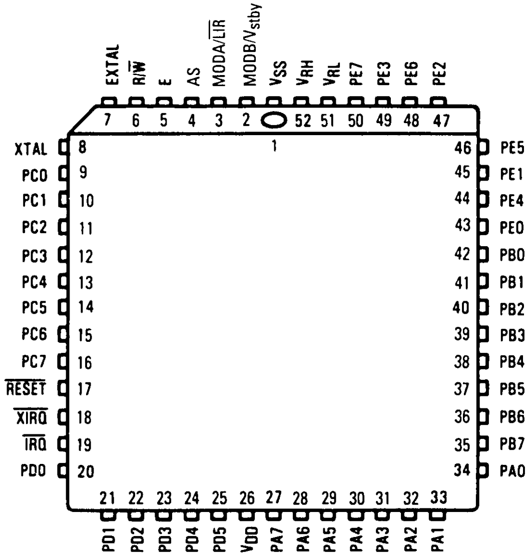
The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC11 series HCMOS microcontroller devices.

Package Type	Temperature	CONFIG	Description	MC Part Number
PLCC (FN Suffix)	−40° to +85°C	$0F	BUFFALO ROM	MC68HC11E9FN
	−40 to +105°C	$0F	BUFFALO ROM	MC68HC11E9VFN1
	−40 to +125°C	$0F	BUFFALO ROM	MC68HC11E9MFN1
	−40 to +85°C	$0D	No ROM	MC68HC11E1FN
	−40 to +105°C	$0D	No ROM	MC68HC11E1VFN
	−40 to +125°C	$0D	No ROM	MC68HC11E1MFN
	−40 to +85°C	$0C	No ROM, No EEPROM	MC68HC11E0FN

MS-™DOS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

52-Lead Quad Package


EXTAL 7
R/$\overline{W}$ 6
E 5
AS 4
MODA/$\overline{LIR}$ 3
MODB/V_{stby} 2
V_{SS} 1
V_{RH} 52
V_{RL} 51
PE7 50
PE3 49
PE6 48
PE2 47
XTAL 8
PC0 9
PC1 10
PC2 11
PC3 12
PC4 13
PC5 14
PC6 15
PC7 16
$\overline{RESET}$ 17
$\overline{XIRQ}$ 18
$\overline{IRQ}$ 19
PD0 20
PD1 21
PD2 22
PD3 23
PD4 24
PD5 25
V_{DD} 26
PA7 27
PA6 28
PA5 29
PA4 30
PA3 31
PA2 32
PA1 33
PA0 34
PB7 35
PB6 36
PB5 37
PB4 38
PB3 39
PB2 40
PB1 41
PB0 42
PE0 43
PE4 44
PE1 45
PE5 46

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC11F1

Product Preview

8-Bit Microcontroller

3

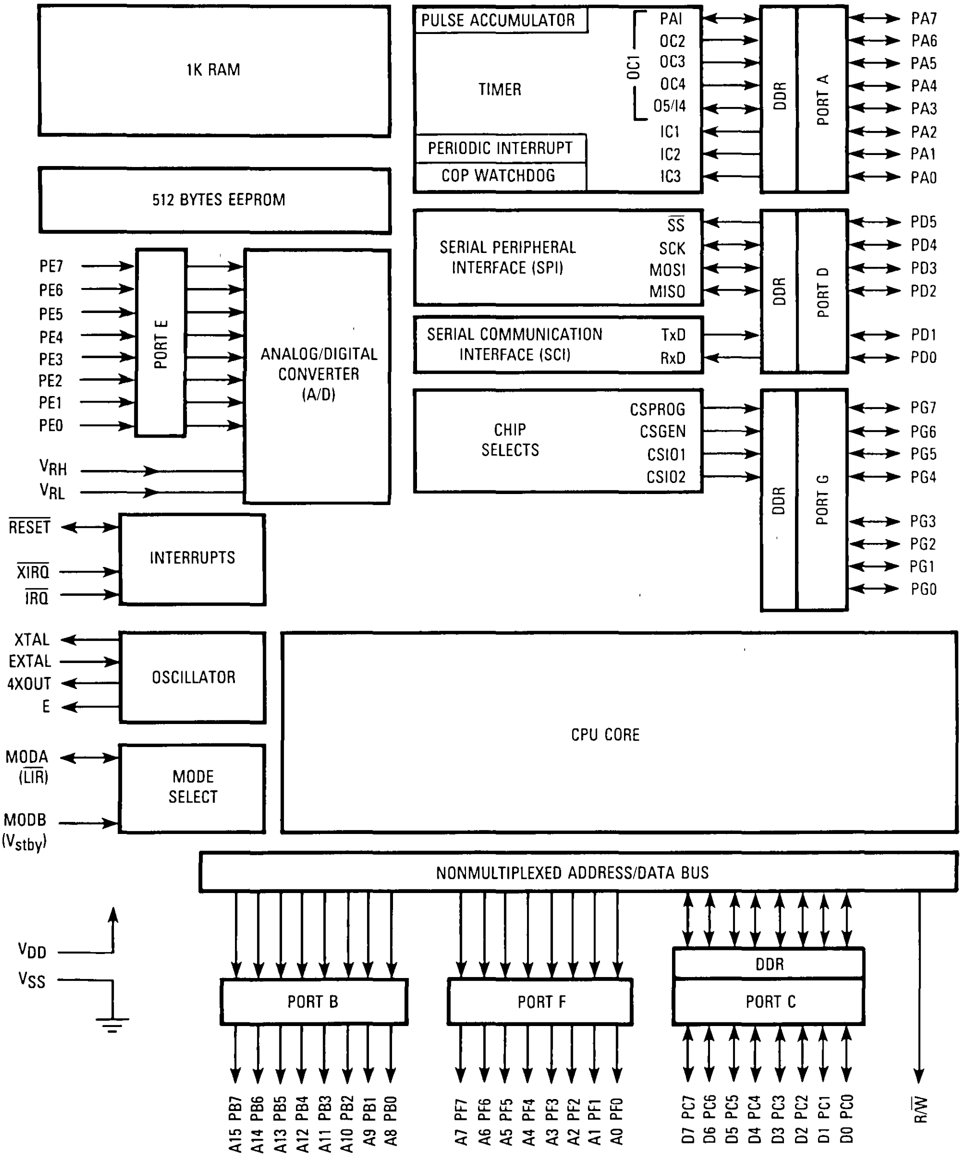
The MC68HC11F1 is an advanced 8-bit microcontroller unit (MCU) with highly sophisticated on-chip peripheral functions. The MC68HC11F1 is a ROMless version of the M68HC11 Family of microcontrollers with several enhancements.

The following are some of the features of the MC68HC11F1:

- HCMOS Technology
- 1024 Bytes of Static RAM (All Saved during Standby)
- 512 Bytes of EEPROM
- Nonmultiplexed Bus for Higher Frequency Operation
- 64K Memory Addressability
- Four Programmable Chip Selects
- Block Protect Mechanism for EEPROM and CONFIG
- Enhanced 16-Bit Timer System
  Four Stage Programmable Prescaler,
  Three Input Capture/Four Output Compare Functions, or
  Four Input Capture/Four Output Compare Functions (Software Selectable)
- 8-Bit Pulse Accumulator Circuit
- Real-Time Interrupt Circuit
- Enhanced NRZ Serial Communication Interface (SCI)
- Serial Peripheral Interface (SPI)
- Eight-Channel 8-Bit A/D Converter
- Computer Operating Properly (COP) Watchdog Timer
- 68-Pin PLCC Package
- Up To Eight Additional I/O Pins from ROMless 52-Pin Version

This document contains informaion on a product under development Motorola reserves the right to change or discontinue this product without notice.

BLOCK DIAGRAM


1K RAM
512 BYTES EEPROM
PULSE ACCUMULATOR
TIMER
PERIODIC INTERRUPT
COP WATCHDOG
OC1
PAI
OC2
OC3
OC4
O5/I4
IC1
IC2
IC3
DDR
PORT A
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
SERIAL PERIPHERAL INTERFACE (SPI)
$\overline{SS}$
SCK
MOSI
MISO
SERIAL COMMUNICATION INTERFACE (SCI)
TxD
RxD
DDR
PORT D
PD5
PD4
PD3
PD2
PD1
PD0
CHIP SELECTS
CSPROG
CSGEN
CSIO1
CSIO2
DDR
PORT G
PG7
PG6
PG5
PG4
PG3
PG2
PG1
PG0
PE7
PE6
PE5
PE4
PE3
PE2
PE1
PE0
PORT E
ANALOG/DIGITAL CONVERTER (A/D)
V_{RH}
V_{RL}
$\overline{RESET}$
$\overline{XIRQ}$
$\overline{IRQ}$
INTERRUPTS
XTAL
EXTAL
4XOUT
E
OSCILLATOR
MODA ($\overline{LIR}$)
MODB (V_{stby})
MODE SELECT
CPU CORE
NONMULTIPLEXED ADDRESS/DATA BUS
V_{DD}
V_{SS}
PORT B
PORT F
DDR
PORT C
A15 PB7
A14 PB6
A13 PB5
A12 PB4
A11 PB3
A10 PB2
A9 PB1
A8 PB0
A7 PF7
A6 PF6
A5 PF5
A4 PF4
A3 PF3
A2 PF2
A1 PF1
A0 PF0
D7 PC7
D6 PC6
D5 PC5
D4 PC4
D3 PC3
D2 PC2
D1 PC1
D0 PC0
R/$\overline{W}$

MC68HC711D3

Product Preview

One Time Programmable ROM (OTPROM) or Standard Eraseable Programmable Read-Only Memory (EPROM) Microcomputer

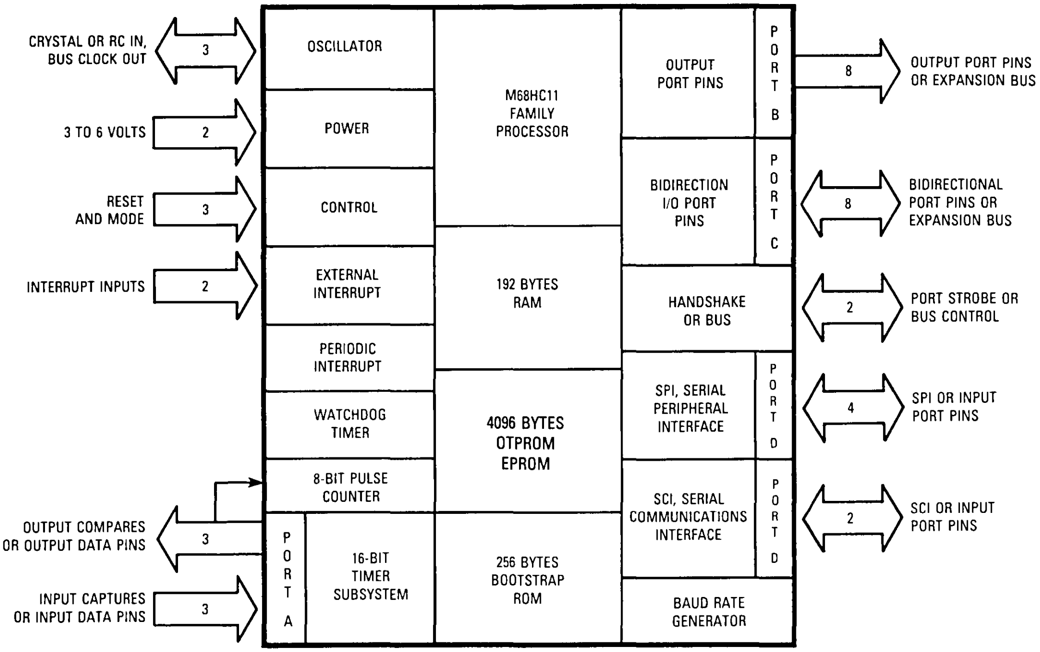
The MC68HC711D3 Microcomputer (MCU) device is similar to the MC68HC11A8 with the following exceptions. The exceptions incorporate 4096 bytes of One Time Programmable Read-Only Memory (OTPROM) or Standard Eraseable Programmable Read-Only Memory (EPROM), 192 bytes of RAM, and no A/D Converter (refer to the block diagram).

The following are some of the hardware and software features of the MC68HC711D3.

- HCMOS Technology
- 4096 Bytes of User OTPROM
- 192 Bytes of RAM
- 256 Bytes of Bootstrap ROM
- Enhanced 16-Bit Timer System:
  - Four Stage Programmable Prescaler
  - Three Input Capture Functions
  - Three Output Compare Functions
- 8-Bit Pulse Accumulator Circuit
- Enhanced NRZ Serial Communications Interface (SCI)
- Serial Peripheral Interface (SPI)
- Real Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog Timer
- Enhanced M6800/M6801 Instruction Set
- 16×16 Integer and Fractional Divide Features
- Bit Manipulation
- WAIT Mode
- STOP Mode
- Expansion Mode Addressable to 64K
- 40-Pin DIP, 44-Pin PLCC (OTPROM) Package
- 40-Pin DIP (EPROM Window) Package

This document contains information on a new product. Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM


CRYSTAL OR RC IN, BUS CLOCK OUT
3
3 TO 6 VOLTS
2
RESET AND MODE
3
INTERRUPT INPUTS
2
OUTPUT COMPARES OR OUTPUT DATA PINS
3
INPUT CAPTURES OR INPUT DATA PINS
3
OSCILLATOR
POWER
CONTROL
EXTERNAL INTERRUPT
PERIODIC INTERRUPT
WATCHDOG TIMER
8-BIT PULSE COUNTER
PORT A
16-BIT TIMER SUBSYSTEM
M68HC11 FAMILY PROCESSOR
192 BYTES RAM
4096 BYTES OTPROM EPROM
256 BYTES BOOTSTRAP ROM
OUTPUT PORT PINS
PORT B
BIDIRECTION I/O PORT PINS
PORT C
HANDSHAKE OR BUS
SPI, SERIAL PERIPHERAL INTERFACE
PORT D
SCI, SERIAL COMMUNICATIONS INTERFACE
PORT D
BAUD RATE GENERATOR
8
OUTPUT PORT PINS OR EXPANSION BUS
8
BIDIRECTIONAL PORT PINS OR EXPANSION BUS
2
PORT STROBE OR BUS CONTROL
4
SPI OR INPUT PORT PINS
2
SCI OR INPUT PORT PINS

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68HC811E2

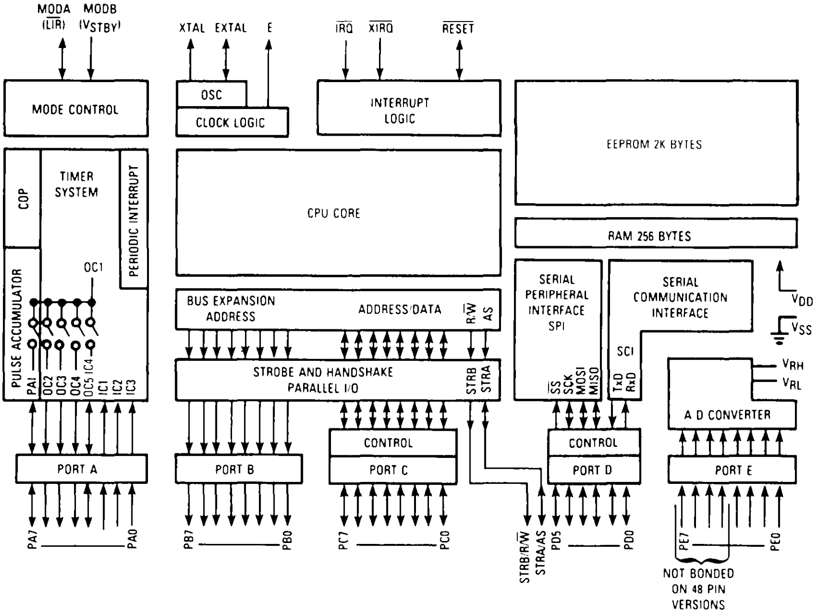
Technical Summary
8-Bit Microcontroller

The MC68HC811E2 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 2K Bytes of EEPROM
- 256 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

BLOCK DIAGRAM



MODA (LIR)
MODB (VSTBY)
XTAL
EXTAL
E
IRQ
XIRQ
RESET
MODE CONTROL
OSC
CLOCK LOGIC
INTERRUPT LOGIC
EEPROM 2K BYTES
COP
TIMER SYSTEM
PERIODIC INTERRUPT
CPU CORE
RAM 256 BYTES
PULSE ACCUMULATOR
OC1
PAI
OC2
OC3
OC4
OC5 IC4
IC1
IC2
IC3
BUS EXPANSION ADDRESS
ADDRESS/DATA
R/W
AS
STROBE AND HANDSHAKE PARALLEL I/O
STRB
STRA
SERIAL PERIPHERAL INTERFACE SPI
SS
SCK
MOSI
MISO
SERIAL COMMUNICATION INTERFACE
SCI
TxD
RxD
VDD
VSS
VRH
VRL
A/D CONVERTER
PORT A
PORT B
CONTROL
PORT C
CONTROL
PORT D
PORT E
PA7
PA0
PB7
PB0
PC7
PC0
STRB/R/W
STRA/AS
PD5
PD0
PE7
PE0
NOT BONDED ON 48 PIN VERSIONS

This document contains information on a new product. Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE (MODE0)

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU.

EXPANDED MULTIPLEXED MODE (MODE1)

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{\text{W}}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits. This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{\text{RESET}}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins

$\overline{\text{IRQ}}$

This pin provides the capability for asynchronously applying interrupts to the MCU Either negative edge-sensitive or level-sensitive triggering is program selectable This pin is configured to level-sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{\text{IRQ}}$

$\overline{\text{XIRQ}}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register is set, and any interrupt is masked until enabled by software This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{\text{LIR}}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{\text{LIR}}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The V_{stby} (voltage standby) is used to retain RAM contents during device powerdown. The mode selections are shown below.

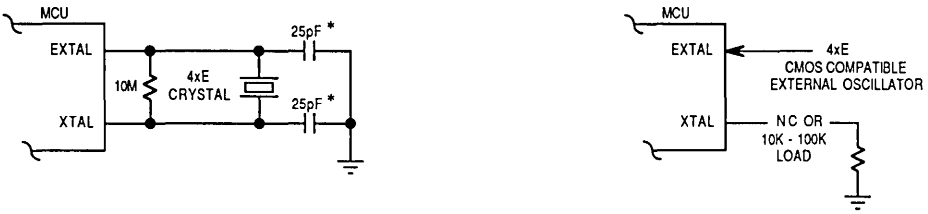
MODB	MODA	MODE SELECTED
1	0	Single Chip
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

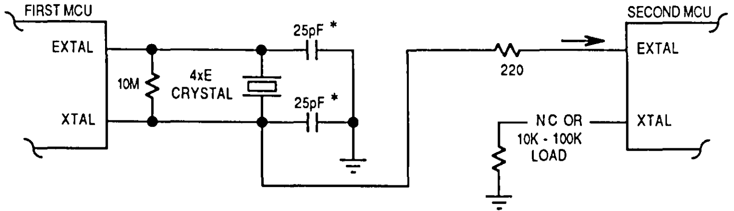
R/$\overline{\text{W}}$/STRB

This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides R/$\overline{\text{W}}$ (read-write) function. The R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.



Common Crystal Connections

External Oscillator Connections



*Includes all stray capacitances

One Crystal Driving Two MCUs

Figure 1. Oscillator Collections

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for four input capture functions and three output compare functions; four output compare functions and three input capture functions; and a pulse a accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/IC4/and-or OC1	PA3/OC5/IC4/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/SS	PD5SS
	STRA	AS
	STRB	R/W
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE3/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4##	PE4/AN4##
E-5	PE5/AN5##	PE5/AN5##
E-6	PE6/AN6##	PE6/AN6##
E-7	PE7/AN7##	PE7/AN7##

##Not Bonded in 48-Pin Versions

of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of parallel I/O where the STRA input and STRB output acts as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/W signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between s shown in Figure 2. In the single-chip mode, the MCU does not (EXT) are for externally addressed memory and I/O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

ACCUMULATOR A AND B

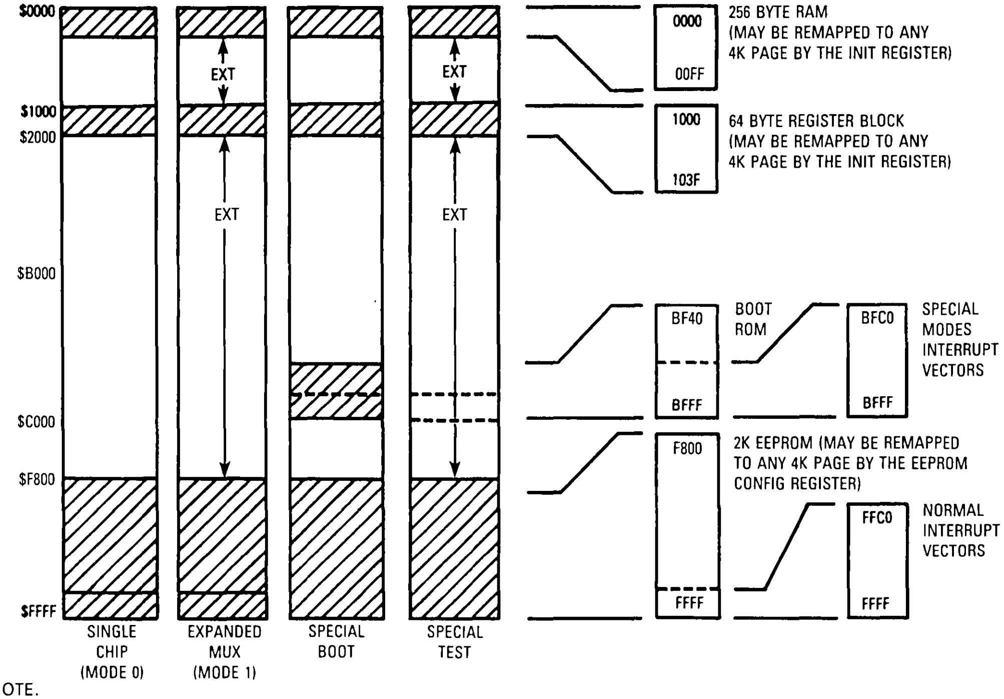
These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.

7	A	0	7	B	0
15		D			0

INDEX REGISTER X (IX)

This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.

15	IX	0


$0000
$1000
$2000
$B000
$C000
$F800
$FFFF
EXT
EXT
EXT
EXT
SINGLE CHIP (MODE 0)
EXPANDED MUX (MODE 1)
SPECIAL BOOT
SPECIAL TEST
0000
00FF
256 BYTE RAM (MAY BE REMAPPED TO ANY 4K PAGE BY THE INIT REGISTER)
1000
103F
64 BYTE REGISTER BLOCK (MAY BE REMAPPED TO ANY 4K PAGE BY THE INIT REGISTER)
BF40
BFFF
BOOT ROM
BFC0
BFFF
SPECIAL MODES INTERRUPT VECTORS
F800
FFFF
2K EEPROM (MAY BE REMAPPED TO ANY 4K PAGE BY THE EEPROM CONFIG REGISTER)
FFC0
FFFF
NORMAL INTERRUPT VECTORS

NOTE.

1. Either or both the internal RAM and registers can be remapped to any 4K boundary by software.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TI4O5	Output Compare 5 Register
$101F	Bit 7	—	—	—	—	—	—	Bit 0		Output Compare 5 Register Input Capture 4 Register
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Reg 1
$1023	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Reg 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Reg 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Reg 2
$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL	Pulse Accum Control Reg
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accum Count Reg
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1

Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035				PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT	EEPROM Block Protect Reg
$1036 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Cir.
$103B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG	EEPROM Prog Control Reg
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Reg
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F	EE3	EE2	EE1	EE0		NOCOP		EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.

15	IY	0

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

15	PC	0

STACK POINTER (SP)

The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the accumulators A and B and registers IX and IY can be stored during certain instructions.

15	SP	0

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

7							0
S	X	H	I	N	Z	V	C

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or $\overline{\text{XIRQ}}$) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

RESETS

The MCU can be reset four ways: 1) an active low input to the $\overline{\text{RESET}}$ pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

RESET PIN

To request an external reset, the $\overline{\text{RESET}}$ pin must be held low for eight E_{cyc} (two E_{cyc} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external $\overline{\text{RESET}}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

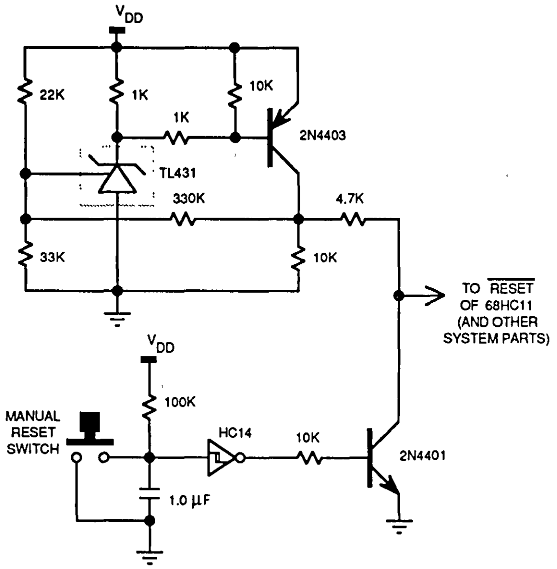
The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system

The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

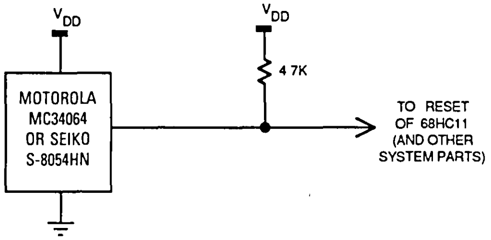
CLOCK MONITOR RESET

The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the $\overline{\text{RESET}}$ pin is driven low to reset the external system.

The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.


VDD
22K
1K
10K
1K
2N4403
TL431
330K
4.7K
33K
10K
TO RESET OF 68HC11 (AND OTHER SYSTEM PARTS)
VDD
100K
MANUAL RESET SWITCH
HC14
10K
2N4401
1.0 µF

Reset Circuit with LVI and RC Delay


VDD
VDD
MOTOROLA MC34064 OR SEIKO S-8054HN
4 7K
TO RESET OF 68HC11 (AND OTHER SYSTEM PARTS)

Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

Table 2. COP Timeout Periods

CR1	CR0	$E/2^{15}$ Divided By	XTAL $= 2^{23}$ Timeout −1/+15.6 ms	XTAL = 8.0 MHz Timeout −0/+16.4 ms	XTAL = 4.9152 MHz Timeout −0/+26.7 ms	XTAL = 4.0 MHz Timeout −0/+32.8 ms	XTAL = 3.6864 MHz Timeout −0/+35.6 ms
0	0	1	15.625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65 536 ms	106 67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

INTERRUPTS

There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the XIRQ pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the XIRQ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

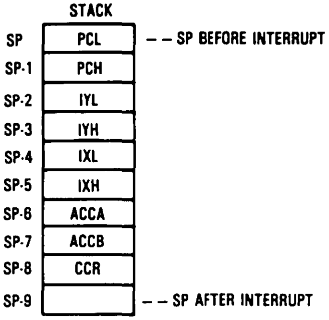


Figure 4. Stacking Order

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
*	*		
*	*		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Input Capture 4/Output Compare 5	I Bit	I4O5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	IRQ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	XIRQ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	RESET	None	None

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be $E/2^{13}$, $E/2^{14}$, $E/2^{15}$, or $E/2^{16}$.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ or $\overline{\text{RESET}}$. An external interrupt used at $\overline{\text{IRQ}}$ is only efective if the I bit in the CCR is clear. An external interrupt applied at the $\overline{\text{XIRQ}}$ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the $\overline{\text{XIRQ}}$ request. If the X bit is set, the processing will always continue with the instruction immediately following the STOP instruction. A low input to the $\overline{\text{RESET}}$ pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

A restart delay is required if the internal oscillator is being used, to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the $\overline{\text{RESET}}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

WAIT

The wait (WAI) instruction places the MCU in a low-power consumption mode, but the wait mode consumes slightly more power than the stop mode. In the wait mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or $\overline{\text{RESET}}$. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the wait mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces wait mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from $FFFF to $0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has four input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are four 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA3 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Port A pin 3 serves multiple functions. After reset, data direction bit 3 (DDRA3), in the PACTL register is cleared to zero configuring port A pin 3 as an input. Port A pin 3 can then be used as a input capture 4 (IC4), by setting I4/O5 to "one" in the PACTL register. The I4/O5 bit is configured to OC5 (cleared to zero) on reset. If DDRA3 is configured as an output and IC4 is enabled, writes to port A bit 3 causes edges on the PA3 to result in input captures. When the TI405 register is acting as the IC4 capture register it cannot be written to. When PA3 is being used as IC4, writes to TI405 register have no meaning.

TIMER CONTROL REGISTER 2 (TCTL2) $1021

	7	6	5	4	3	2	1	0
	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET	0	0	0	0	0	0	0	0

EDGxB and EDGxA — Input Capture x Edge Control

These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to $FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one. Upon reset, I4/O5 is configured as OC5. The OC5 function overrides DDRA3 to force the Port A pin 3 to be an output whenever OM5:OL5 bits are not 0:0. In all other aspects, OC5 works the same as the other output compares.

TIMER COMPARE FORCE REGISTER (CFORC) $100B

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

	7	6	5	4	3	2	1	0
	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET	0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action
1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set
0 = Has no meaning

Bits 2-0 — Not Implemented
These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M) $100C

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D) $100D

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

	7	6	5	4	3	2	1	0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1) $1020

	7	6	5	4	3	2	1	0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET	0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode
OL2-OL5 — Output Level
These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TIMER INTERRUPT MASK REGISTER 1 (TMSK1)

	7	6	5	4	3	2	1	0
	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I
RESET	0	0	0	0	0	0	0	0

OCxI — Output Compare x Interrupt
1 = Interrupt sequence requested if OCxF = 1 in TFLG1
0 = Interrupt inhibited

ICxI — Input Capture x Interrupt
1 = Interrupt sequence requested if ICxF = 1 in TFLG1
0 = Interrupt inhibited

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5I bit behaves as the input capture 4 interrupt bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 interrupt control bit.

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position.

	7	6	5	4	3	2	1	0
	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F
RESET	0	0	0	0	0	0	0	0

OCxF — Output Compare x Flag

Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).

1 = Bit cleared
0 = Not affected

ICxF — Input Capture x Flag

Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).

1 = Bit cleared
0 = Not affected

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5F bit behaves as the input capture 4 flag bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 flag.

TIMER INTERRUPT MASK REGISTER 2 (TMSK2) $1024

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG1. Two timer prescaler bits are also included in this register.

	7	6	5	4	3	2	1	0
	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable

1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled

RTII — RTI Interrupt Enable

1 = Interrupt requested when RTIF = 1
0 = RTIF interrupt disabled

PAOVI — Pulse Accumulator Overflow Interrupt Enable

1 = Interrupt requested when PAOVF = 1
0 = PAOVF disabled

PAII — Pulse Accumulator Input Interrupt Enable

1 = Interrupt requested when PAIF = 1
0 = PAIF disabled

Bits 3-2 — Not Implemented

These bits always read zero.

PR1 and PR0 — Timer Prescaler Selects

Can only be written to during initialization. Writes are disabled after the first write or after 64 E cycles out of reset.

PR1	PR0	Divide-by-Factor
0	0	1
0	1	4
1	0	8
1	1	16

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2) $1025

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

	7	6	5	4	3	2	1	0
	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET	0	0	0	0	0	0	0	0

TOF — Timer Overflow

Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag

Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag

Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag

Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented

These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0
	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7

1 = Output
0 = Input only

PAEN — Pulse-Accumulator System Enable

1 = Pulse accumulator on
0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode

1 = Gated time accumulator
0 = External even counting

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6 67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

PEDGE — Pulse Accumulator Edge Control

This bit provides clock action along with PAMOD.

1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.

0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

DDRA3 — Data Directional for Port A Bit 3

1 = Output

0 = Input only

I4/O5 — Input 4/Output 5

1 = Input capture 4 function enabled (No OC5)

0 = Output compare 5 function enabled (No IC4)

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

EEPROM PROGRAMMING

The 2K bytes of EEPROM are located at $F800 through $FFC0. Programming of the EEPROM is controlled by the EEPROM programming control register (PPROG). The EEPROM is disabled when the EEON bit in the system configuration register (CONFIG) is zero. Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz, the efficiency of this charge pump decreases, which increases the time required to program or erase a location. Recommended program and erase time is 10 milliseconds when the E clock is between 2 MHz and should be increased to as much as 20 milliseconds when E clock is between 1 MHz and 2 MHz. When E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. The following paragraphs describe how to program or erase the EEPROM using the PPROG control register.

EEPROM BLOCK PROTECT REGISTER (BPROT) $1035

This 5-bit register protects against inadvertent writes to the CONFIG register and to the EEPROM. To permit the user to separate EEPROM into categories like 'temporary' or 'permanent', EEPROM is divided into four individually protected blocks. The CONFIG register is also protected.

In normal operating modes, EEPROM and CONFIG are protected out of reset, and the user has 64 E clock cycles to unprotect any of the blocks that will require programming or erasing. The BPROT register bits can only be cleared, written to zero, during the first 64 E clock cycles after reset. Once the bits are cleared, the associated EEPROM section and/or the CONFIG register can be programmed or erased in the normal manner. The EEPROM is visible only if the EEON bit in the CONFIG register is set. In the test or bootstrap modes, bits of the BPROT register can be set or cleared at any time. In either single-chip or expanded mode, BPROT register bits can be written back to one anytime after the first 64 E clock cycles in order to protect the EEPROM and/or the CONFIG register. However, these bits can only be cleared again in the test or bootstrap modes.

	7	6	5	4	3	2	1	0
	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET	0	0	0	1	1	1	1	1

Bits 7–5 — Not Implemented

These bits always read zero

PTCON — Protect CONFIG Register Bit

1 = Programming/erasure of the CONFIG register disabled

0 = Programming/erasure of the CONFIG register allowed

BPRT3–BPRT0 — Block Protect Bits

1 = A set bit protects a block of EEPROM against programming or erasure.

0 = A cleared bit permits programming or erasure of the associated block.

Bit	Block Protected	Block Size
BPRT0	$1800–19FF	512 Bytes
BPRT1	$1A00–1BFF	512 Bytes
BPRT2	$1C00–1DFF	512 Bytes
BPRT3	$1E00–1FFF	512 Bytes

ERASING THE EEPROM

Erasure of the EEPROM is controled by bit settings in PPROG, and the appropriate bits in the BPROT register must also be cleared before the EEPROM can be changed. Programs can be written to perform bulk, row, or byte erase. In bulk erase, all 512 bytes of the EEPROM are erased. In row erase, 16 bytes ($B600-$B60F, $B610-$B61F), etc) are erased. Other MCU operations can continue to be performed during erasing provided the operations do not include reads of data from EEPROM.

PROGRAMMING EEPROM

During programming, the ROW and BYTE bits are not used. If the E clock frequency is 1 MHz or less, the CSEL bit in the OPTION register must be set. Zeros must be erased by a separate erase operation before programming. Other MCU operations can continue to be performed during programming provided the operations do not include reads of data from EEPROM.

EEPROM PROGRAMMING CONTROL REGISTER (PPROG) $103B

This 8-bit register is used to control programming and erasure of the EEPROM. This register is cleared on reset so the EEPROM is configured for normal reads.

	7	6	5	4	3	2	1	0
	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET	0	0	0	0	0	0	0	0

ODD — Program Odd Rows (TEST)
EVEN — Program Even Rows (TEST)
Bit 5 — Not Implemented
This bit always reads zero.
BYTE — Byte Erase Select
This bit overrides the ROW bit.
1 = Erase only one byte
0 = Row or bulk erase
ROW — Row Erase Select
If BYTE bit = 1, ROW has no meaning.
1 = Row erase
0 = Bulk or byte erase
ERASE — Erase Mode Select
1 = Erase mode
0 = Normal read or program
EELAT — EEPROM Latch Control
1 = EEPROM Address and data configured for programmming/erasing
0 = EEPROM Address and data configured for read mode
EEPGM — EEPROM Programming Voltage Enable
1 = Programming voltage turned on
0 = Programming voltage turned off

NOTE

A strict register access sequence must be followed to allow successful programming and erase operations. The following procedures for modifying the EEPROM and CONFIG register detail the sequence. If an attempt is made to set both the EELAT and EEPGM bits in the same write cycle and if this attempt occurs before the required write cycle with the EELAT bit set, then neither is set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation in progress is not disturbed. If no EEPROM address is written between when EELAT is set and EEPGM is set, then no program or erase operation takes place. These safeguards were included to prevent accidental EEPROM changes in cases of program runaway.

ERASING THE CONFIG REGISTER

Erasing the CONFIG register follows the same procedures as that used for the EEPROM including bulk, byte, and row erase. The CONFIG register may be programmed or erased while the MCU is operating in any mode depending on the setting of bit A in BPROT. The bulk erase restriction on CONFIG is not present on all derivatives in the M68HC11 Family. Please check the applicable data sheet or technical summary for the restrictions.

PROGRAMMING THE CONFIG REGISTER

Programming the CONFIG register follows the same procedures as that used for the EEPROM except the CONFIG register address is used. On mask set B96D, the CONFIG register may only be programmed while the MCU is operating in the test or bootstrap mode.

SYSTEM CONFIGURATION REGISTER (CONFIG) $103F

The CONFIG is implemented in EEPROM cells and controls the presence of ROM and EEPROM in the memory map and enables the COP watchdog system.

7	6	5	4	3	2	1	0
EE3	EE2	EE1	EE0	0	NOCOP	0	EEON

EE0-EE3 — EEPROM Map Position
These four bits specify the upper four bits of the EEPROM address. These bit have no meaning in the single-chip mode, because the 2K EEPROM is forced on at locations $F800 through $FFFF.

EE3	EE2	EE1	EE0	Location
0	0	0	0	$0800–$0FFF
0	0	0	1	$1800–$1FFF
0	0	1	0	$2800–$2FFF
0	0	1	1	$3800–$3FFF
0	1	0	0	$4800–$4FFF
0	1	0	1	$5800–$5FFF
0	1	1	0	$6800–$6FFF
0	1	1	1	$7800–$7FFF
1	0	0	0	$8800–$8FFF
1	0	0	1	$9800–$9FFF
1	0	1	0	$A800–$AFFF
1	0	1	1	$B800–$BFFF
1	1	0	0	$C800–$CFFF
1	1	0	1	$D800–$DFFF
1	1	1	0	$E800–$EFFF
1	1	1	1	$F800–$FFFF

Bit 3 — Not Implemented
This bit always reads zero
NOCOP — COP System Disable
1 = COP watchdog system disable
0 = COP watchdog system enabled
Bit 1 — Not Implemented
This bit always reads zero
EEON — Enable On-Chip EEPROM
When this bit is programmed to "zero", the 512-byte EEPROM is disabled, and that memory space becomes externally accessed space.

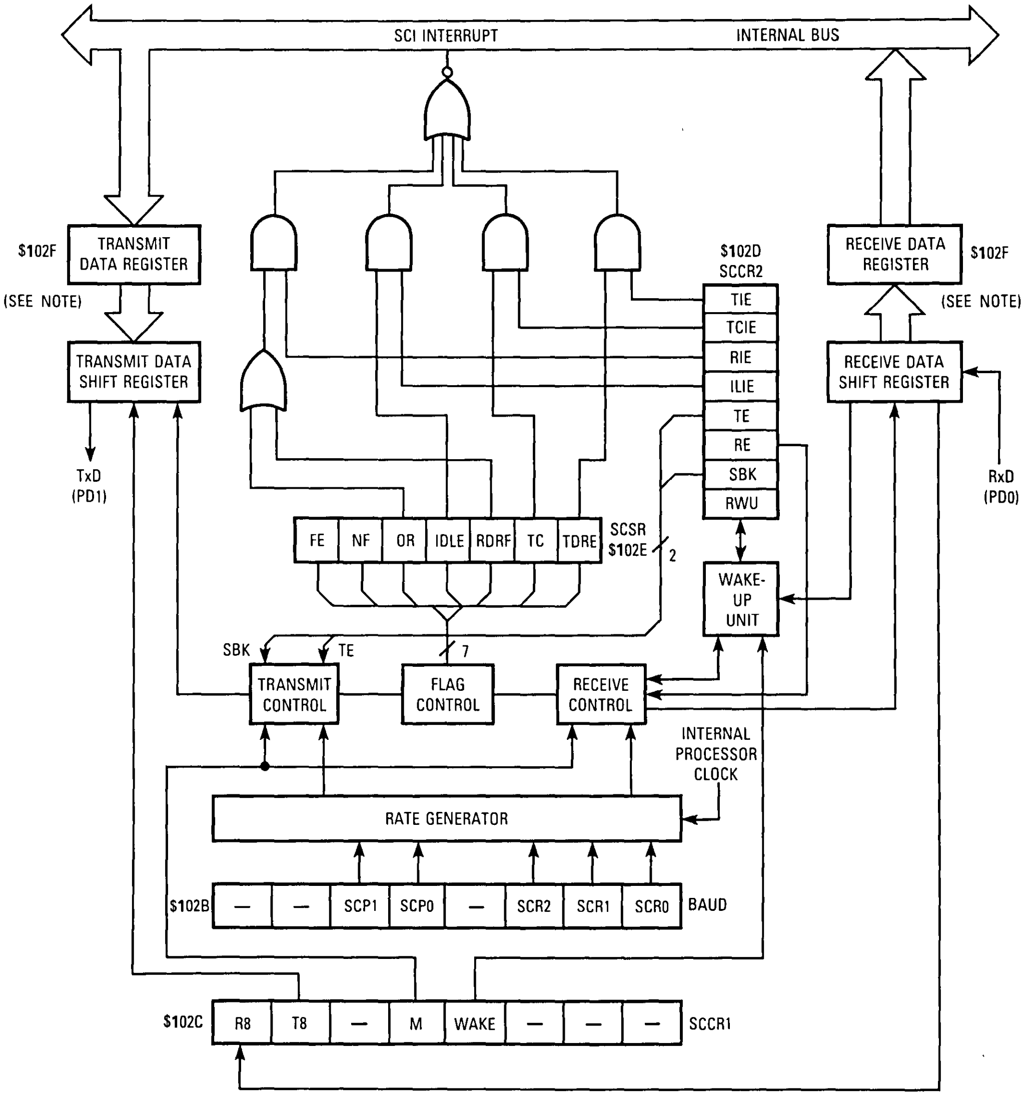
SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be interfaced efficiently with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins PD0 for receive data (RxD), and PD1 for transmit data (TxD). The baud rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/reception of a message;


SCI INTERRUPT
INTERNAL BUS
$102F
TRANSMIT DATA REGISTER
(SEE NOTE)
TRANSMIT DATA SHIFT REGISTER
TxD (PD1)
$102D
SCCR2
TIE
TCIE
RIE
ILIE
TE
RE
SBK
RWU
RECEIVE DATA REGISTER
$102F
(SEE NOTE)
RECEIVE DATA SHIFT REGISTER
RxD (PD0)
FE
NF
OR
IDLE
RDRF
TC
TDRE
SCSR
$102E
2
WAKE-UP UNIT
SBK
TE
7
TRANSMIT CONTROL
FLAG CONTROL
RECEIVE CONTROL
INTERNAL PROCESSOR CLOCK
RATE GENERATOR
$102B
—
—
SCP1
SCP0
—
SCR2
SCR1
SCR0
BAUD
$102C
R8
T8
—
M
WAKE
—
—
—
SCCR1

NOTE: The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read.

Figure 5. SCI Block Diagram

2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR)

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1)

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

	7	6	5	4	3	2	1	0
	R8	T8	0	M	WAKE	0	0	0
RESET	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.

T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.

Bit 5 — Not Implemented
This bit always reads zero.

M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line

Bits 2-0 — Not Implemented
These bits always read zero.

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt if TDRE = 1
0 = TDR interrupts disabled

TCIE — Transmit-Complete Interrupt Enable
1 = SCI interrupt if TC = 1
0 = TC interrupts disabled

RIE — Receive Interrupt Enable
1 = SCI interrupt if RDRF or OR = 1
0 = RDRF or OR interrupt disabled

ILIE — Idle-Line Interrupt Enable
1 = SCI interrupt if IDLE = 1
0 = IDLE interrupts disabled

TE — Transmit Enable
1 = Transmit shift register output is applied to the TxD line
0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
1 = Receiver enabled
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited

RWU — Receiver Wake Up
When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the

WAKE bit is zero, RWU is cleared by the SCI logic after receiving 10 (M=0) or 11 (M=1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If this bit is toggled set and cleared, the transmitter sends 10 (M=0) or 11 (M=1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET							
1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty

1 = Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register

0 = Cleared by a read of SCSR (with TDRE = 1) followed by a write to SCDR

TC — Transmit Complete

1 = Automatically set when all data frame, preamble, or break condition transmissions are complete

0 = Cleared by a read of SCSR (with TC = 1) followed by a write to SCDR

RDRF — Receive Data Register Full

1 = Automatically set when a character is transferred from the receiver shift register to the SCDR

0 = Cleared by a read of SCSR (with RDRF = 1) followed by a read of SCDR

IDLE — Idle-Line Detect

This bit is inhibited while RWU = 1.

1 = Automatically set when the receiver serial input becomes idle after having been active

0 = Cleared by a read of SCSR (with IDLE = 1) followed by a read of SCDR

OR — Overrun Error

1 = Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read

0 = Cleared by a read of SCSR (with OR = 1) followed by a read of SCDR

NF — Noise Flag

1 = Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame

0 = Cleared by a read of SCSR (with NF = 1) followed by a write to SCDR

FE — Framing Error

1 = Automatically set when a logic 0 is detected where a stop bit was expected

0 = Cleared by a read of SCSR (with FE = 1) followed by a read of SCDR

Bit 0 — Not Implemented

This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

7	6	5	4	3	2	1	0
TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET							
0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)

This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.

Bit 6 — Not Implemented

This bit always reads zero.

SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects

These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)

This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

SCR2-SCR0 — SCI Baud-Rate Selects

These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	**1**	**0**		**131.072 K Baud**	**32.768 K Baud**	**76.80 K Baud**	**19.20 K Baud**	**9600 Baud**
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{\text{SS}}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.

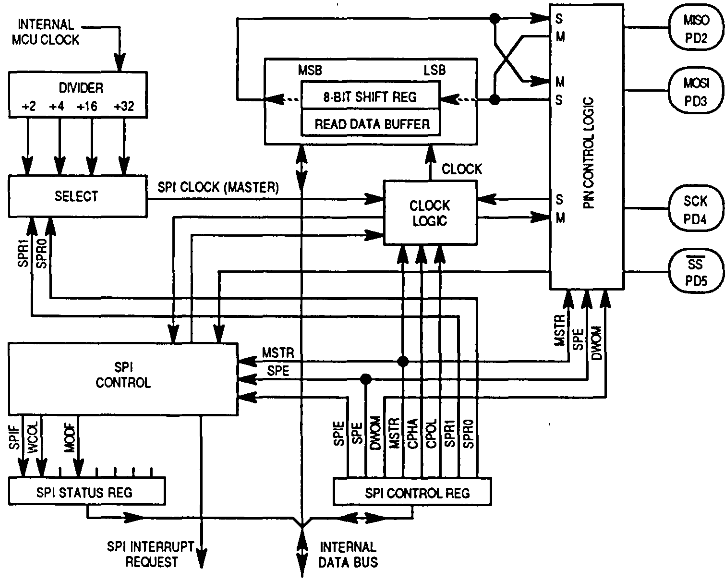

INTERNAL MCU CLOCK
DIVIDER
÷2 ÷4 ÷16 ÷32
SELECT
SPR1
SPR0
SPI CLOCK (MASTER)
MSB
LSB
8-BIT SHIFT REG
READ DATA BUFFER
CLOCK
CLOCK LOGIC
S
M
M
S
S
M
PIN CONTROL LOGIC
MISO PD2
MOSI PD3
SCK PD4
SS PD5
MSTR
SPE
DWOM
SPI CONTROL
MSTR
SPE
SPIF
WCOL
MODF
SPI STATUS REG
SPIE
SPE
DWOM
MSTR
CPHA
CPOL
SPR1
SPR0
SPI CONTROL REG
SPI INTERRUPT REQUEST
INTERNAL DATA BUS

Figure 6. SPI Block Diagram

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR) $1028

7	6	5	4	3	2	1	0
SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET							
0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
- 1 = SPI interrupt if SPIF = 1
- 0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable
- 1 = SPI system on
- 0 = SPI system off

DWOM — Port D Wire-OR Mode Option

This bit affects all six port D pins together.
- 1 = Port D outputs act as open-drain outputs
- 0 = Port D outputs are normal CMOS outputs

MSTR — Master Mode Select
- 1 = Master mode
- 0 = Slave mode

CPOL — Clock Polarity

This bit selects the polarity of the SCK clock.
- 1 = SCK line idles high
- 0 = SCK line idles low

CPHA — Clock Phase

This bit selects one of two fundamentally different clock protocols. Refer to Figure 7. If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{SS}$ is low and ends when the SPIF flag gets set.

SPR1 and SPR0 — SPI Clock Rate Select

These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR) $1029

7	6	5	4	3	2	1	0
SPIF	WCOL	0	MODF	0	0	0	0
RESET							
0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag
- 1 = Automatically set when data transfer is complete between processor and external device
- 0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision
- 1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred

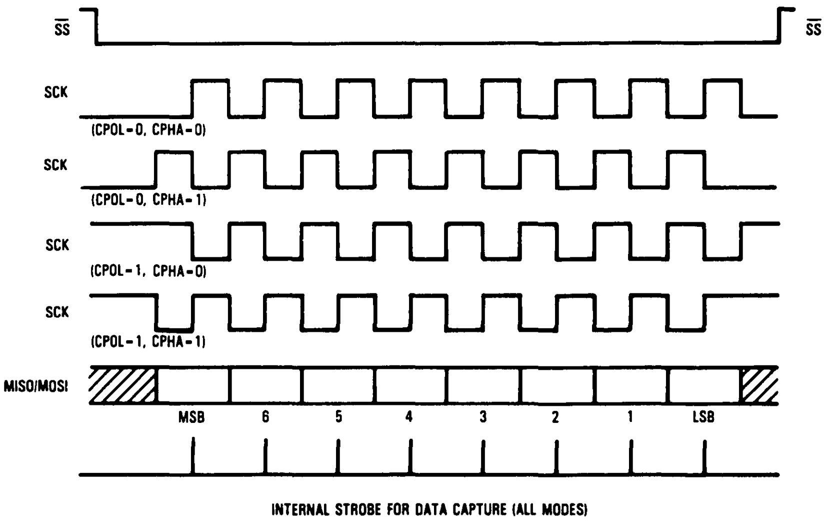


Figure 7. Data Clock Timing Diagram

0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented

This bit always reads zero.

MODF — Mode Fault

This bit indicates the possibility of a multi-master conflict for system control and therefore allows a proper exit from system operation to a reset or default system state.

1 = Automatically set when a master device has its $\overline{SS}$ pin pulled low

0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented

These bits always read zero.

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.
2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB

— Continued—

Function	Mnemonic
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16×16	FDIV
Integer Divide 16×16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8×8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY

— Continued —

Function	Mnemonic
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if=zero	BEQ
Branch if≥zero	BGE
Branch if)zero	BGT
Branch if Higher	BHI
Branch if≤Zero	BLE
Branch if Lower or Same	BLS
Branch if<Zero	BLT
Branch if Minus	BMI
Branch if not=Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

OPCODE MAP SUMMARY

Table 6 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

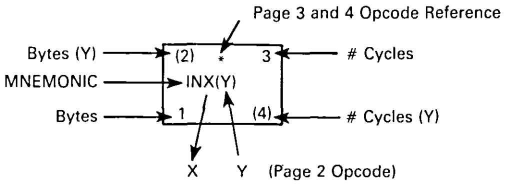
Table 6. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
HI LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI LOW
0 0000	TEST 1	2 SBA 1	3 BRA 2	(2) 3 TSX(Y) 1 (4)	2 NEGA 1	2 NEGB 1	(3) 6 NEG 2 (7)	6 NEG 3	2 SUBA 2	3 SUBA 2	(3) 4 SUBA 2 (5)	4 SUBA 3	2 SUBB 2	3 SUBB 2	(3) 4 SUBB 2 (5)	4 SUBB 3	0
1 0001	2 NOP 1	2 CBA 1	3 BRN 2	3 INS 1					2 CMPA 2	3 CMPA 2	(3) 4 CMPA 2 (5)	4 CMPA 3	2 CMPB 2	3 CMPB 2	(3) 4 CMPB 2 (5)	4 CMPB 3	1
2 0010	41 IDIV 1	6 BRSET 4	3 BHI 2	4 PULA 1					2 SBCA 2	3 SBCA 2	(3) 4 SBCA 2 (5)	4 SBCA 3	2 SBCB 2	3 SBCB 2	(3) 4 SBCB 2 (5)	4 SBCB 3	2
3 0011	41 FDIV 1	6 BRCLR 4	3 BLS 2	4 PULB 1	2 COMA 1	2 COMB 1	(3) 6 COM 2 (7)	6 COM 3	* 4 SUBD 3	* 5 SUBD 2	(3) * 6 SUBD 2 (7)	* 6 SUBD 3	4 ADDD 3	5 ADDD 2	(3) 6 ADDD 2 (7)	6 ADDD 3	3
4 0100	3 LSRD 1	6 BSET 3	3 (BHS) BCC 2	3 DES 1	2 LSRA 1	2 LSRB 1	(3) 6 LSR 2 (7)	6 LSR 3	2 ANDA 2	3 ANDA 2	(3) 4 ANDA 2 (5)	4 ANDA 3	2 ANDB 2	3 ANDB 2	(3) 4 ANDB 2 (5)	4 ANDB 3	4
5 0101	3 (LSLD) ASLD 1	6 BCLR 3	3 (BLO) BCS 2	(2) 3 TX(Y)S 1 (4)					2 BITA 2	3 BITA 2	(3) 4 BITA 2 (5)	4 BITA 3	2 BITB 2	3 BITB 2	(3) 4 BITB 2 (5)	4 BITB 3	5
6 0110	2 TAP 1	2 TAB 1	3 BNE 2	3 PSHA 1	2 RORA 1	2 RORB 1	(3) 6 ROR 2 (7)	6 ROR 3	2 LDAA 2	3 LDAA 2	(3) 4 LDAA 2 (5)	4 LDAA 3	2 LDBB 2	3 LDBB 2	(3) 4 LDBB 2 (5)	4 LDBB 3	6
7 0111	2 TPA 1	2 TBA 1	3 BEQ 2	3 PSHB 1	2 ASRA 1	2 ASRB 1	(3) 6 ASR 2 (7)	6 ASR 3		3 STAA 2	(3) 4 STAA 2 (5)	4 STAA 3		3 STBB 2	(3) 4 STBB 2 (5)	4 STBB 3	7
8 1000	(2) (3) INX(Y) (1) (4)	PAGE 2	3 BVC 2	(2) 5 PULX(Y) 1 (6)	2 ASLA 1	2 ASLB 1	(3) 6 ASL 2 (7)	6 ASL 3	2 EORA 2	(3) 3 EORA 2 (5)	4 EORA 2	4 EORA 3	2 EORB 2	3 EORB 2	(3) 4 EORB 2 (5)	4 EORB 3	8
9 1001	(2) 3 DEX(Y) 1 (4)	2 DAA 1	7 BVS 2	5 RTS 1	2 ROLA 1	2 ROLB 1	(3) 6 ROL 2 (7)	6 ROL 3	2 ADCA 2	3 ADCA 2	(3) 4 ADCA 2 (5)	4 ADCA 3	2 ADCB 2	3 ADCB 2	(3) 4 ADCB 2 (5)	4 ADCB 3	9
A 1010	2 CLV 1	PAGE 3	3 BPL 2	(2) 3 ABX(Y) 1 (4)	2 DECA 1	2 DECB 1	(3) 6 DEC 2 (7)	6 DEC 3	2 DRAA 2	3 DRAA 2	(3) 4 ORAA 2 (5)	4 ORAA 3	2 ORAB 2	3 ORAB 2	(3) 4 ORAB 2 (5)	4 ORAB 3	A
B 1011	2 SEV 1	2 ABA 1	3 BMI 2	12 RTI 1					2 ADDA 2	3 ADDA 2	(3) 4 ADDA 2 (5)	4 ADDA 3	2 ADDB 2	3 ADDB 2	(3) 4 ADDB 2 (5)	4 ADDB 3	B
C 1100	2 CLC 1	(4) 7 BSET 3 (8)	3 BGE 2	(2) 4 PSHX(Y) 1 (5)	2 INCA 1	2 INCB 1	(3) 6 INC 2 (7)	6 INC 3	4 CPX(Y) 3	5 CPX(Y) 2	(3) * 6 CPX(Y) 2 (7)	6 CPX(Y) 3	3 LDD 3	4 LDD 2	(3) 5 LDD 2 (6)	5 LDD 3	C
D 1101	2 SEC 1	(4) 7 BCLR 3 (8)	3 BLT 2	10 MUL 1	2 TSTA 1	2 TSTB 1	(3) 6 TST 2 (7)	6 TST 3	6 BSR 2	5 JSR 2	(3) 6 JSR 2 (7)	6 JSR 3	PAGE 4	4 STD 2	(3) 5 STD 2 (6)	5 STD 3	D
E 1110	2 CLI 1	(5) 7 BRSET 4 (8)	3 BGT 2	12 WAI 1			(3) 3 JMP 2 (4)	3 JMP 2	3 LDS 3	4 LDS 2	(3) 5 LDS 2 (6)	5 LDS 3	3 LDX(Y) 3	4 LDX(Y) 2	(3) 5 LDX(Y) 2 (6)	* 5 LDX(Y) 3	E
F 1111	2 SEI 1	(5) 7 BRCLR 4 (8)	3 BLE 2	14 SWI 1	2 CLRA 1	2 CLRB 1	(3) 6 CLR 2 (7)	6 CLR 3	(2) 3 XGDX(Y) 1 (4)	4 STS 2	(3) 5 STS 2 (6)	5 STS 3	2 STOP 1	(3) 4 STX(Y) 2 (5)	(3) 5 STX(Y) 2 (6)	(4) * 5 STX(Y) 3 (6)	F

*Page 3 and 4 Opcode Reference

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC811E2 MC68HC811E2V MC68HC811E2M	T_A	T_L to T_H −40 to 85 −40 to 105 −40 to 125	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e g , either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC)	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

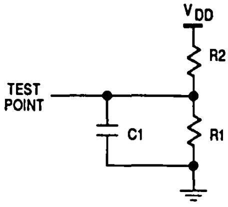
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage All Outputs $I_{Load}=\pm 10.0$ µA (see Note 1) All Outputs Except $\overline{\text{RESET}}$ and MODA	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Output High Voltage All Outputs Except $\overline{\text{RESET}}$, $I_{Load}=-0.8$ mA, $V_{DD}=4.5$ V (see Note 1) XTAL, and MODA	V_{OH}	$V_{DD}-0.8$	—	V
Output Low Voltage All Outputs Except XTAL $I_{Load}=1.6$ mA	V_{OL}	—	0.4	V
Input High Voltage All Inputs Except $\overline{\text{RESET}}$ $\overline{\text{RESET}}$	V_{IH}	$0.7\times V_{DD}$ $0.8\times V_{DD}$	V_{DD} V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	$0.2\times V_{DD}$	V
I/O Ports, Three-State Leakage PA3, PA7, PC0-PC7, PD0-PD5, $V_{in}=V_{IH}$ or V_{IL} AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	I_{OZ}	—	±10	µA
Input Current (see Note 2) $V_{in}=V_{DD}$ or V_{SS} PA0-PA3, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ $V_{in}=V_{DD}$ or V_{SS} MODB/V_{STBY}	I_{in}	 — —	 ±1 ±10	µA
RAM Standby Voltage Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current Powerdown	I_{SB}	—	20	µA
Total Supply Current (see Note 3) RUN: Single Chip Expanded Multiplexed WAIT: All Peripheral Functions Shut Down Single-Chip Mode Expanded Multiplexed Mode STOP: No Clocks, Single-Chip Mode	 I_{DD} W_{IDD} S_{IDD}	 — — — — —	 15 27 6 10 100	 mA mA mA mA µA
Input Capacitance PA0-PA3, PE0-PE7, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	C_{in}	— —	8 12	pF
Power Dissipation Single-Chip Mode Expanded-Multiplexed Mode	P_D	— —	85 150	mW

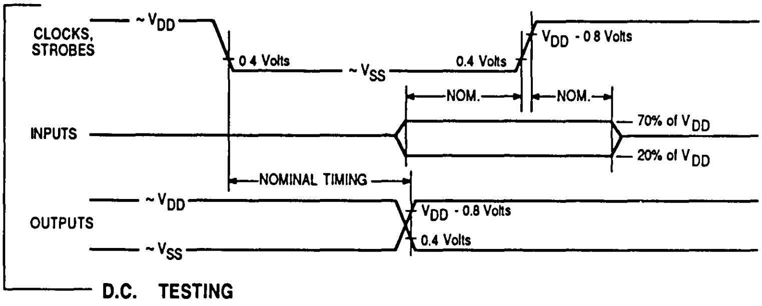
NOTES:

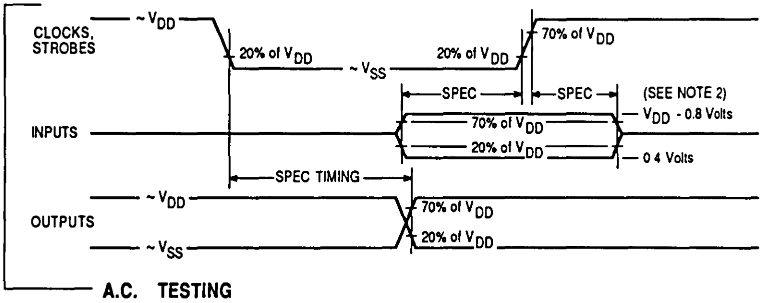
1. V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL}\leq 0.2$ V,
   $V_{IH}\geq V_{DD}-0.2$ V,
   No dc loads,
   EXTAL is driven with a square wave, and
   $t_{cyc}=476.5$ ns.


VDD
R2
TEST POINT
C1
R1

Equivalent Test Load[1]

Pins	R1	R2	C1
PA3-PA7 PB0-PB7 PC0-PC7 PD0, PD5 E, AS, R/W	3.26K	2.38K	90pF
PD1-PD4	3.26K	2.38K	200pF


CLOCKS, STROBES
~VDD
0.4 Volts
~VSS
0.4 Volts
VDD - 0.8 Volts
NOM.
NOM.
INPUTS
70% of VDD
20% of VDD
NOMINAL TIMING
OUTPUTS
~VDD
VDD - 0.8 Volts
~VSS
0.4 Volts
D.C. TESTING


CLOCKS, STROBES
~VDD
20% of VDD
~VSS
20% of VDD
70% of VDD
SPEC
SPEC
(SEE NOTE 2)
INPUTS
70% of VDD
20% of VDD
VDD - 0.8 Volts
0.4 Volts
SPEC TIMING
OUTPUTS
~VDD
70% of VDD
~VSS
20% of VDD
A.C. TESTING

NOTES:

1. Full test loads are applied during all ac electrical test and ac timing measurements
2. During ac timing measurements, inputs are driven to 0.4 volts and $V_{DD}-0.8$ volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

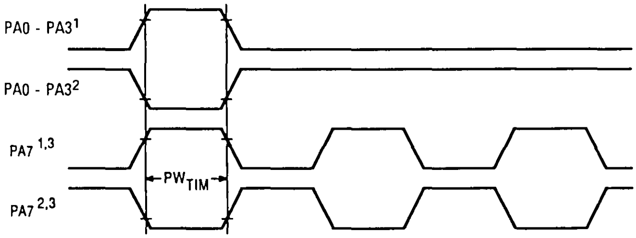
Figure 8. Test Methods

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	$4\ f_o$	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and 13) $t_{PCS} = 1/4\ t_{cyc} - 50$ ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time; May be Preempted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) $PW_{IRQ} = t_{cyc} + 20$ ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) $PW_{TIM} = t_{cyc} + 20$ ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:

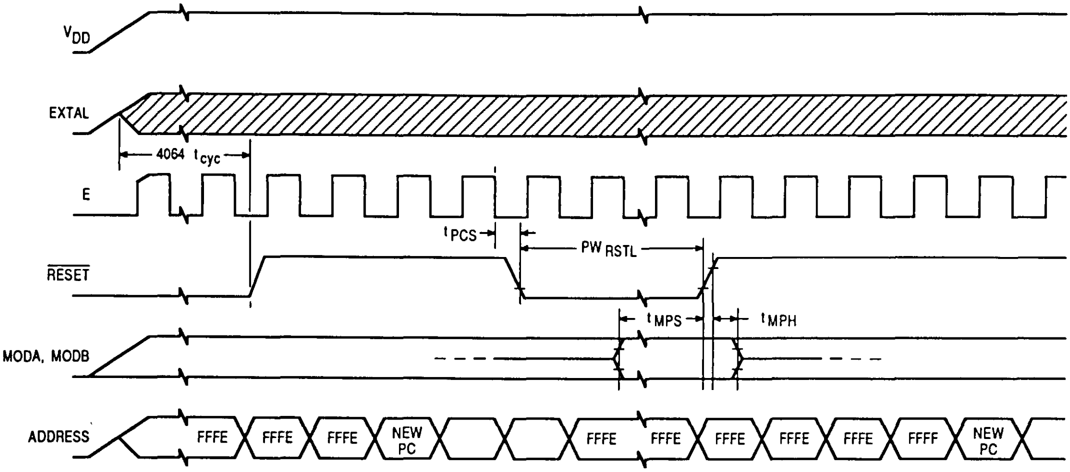
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



NOTES.

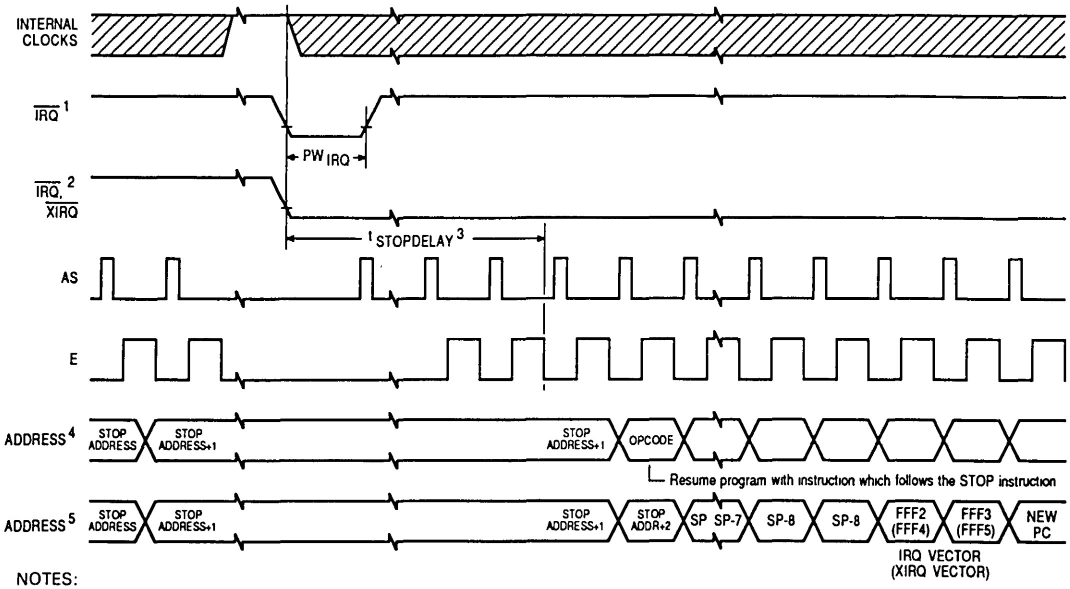
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2

Figure 9. Timer Inputs Timing Diagram


VDD
EXTAL
4064 tcyc
E
tPCS
PW RSTL
RESET
tMPS
tMPH
MODA, MODB
ADDRESS
FFFE
FFFE
FFFE
NEW PC
FFFE
FFFE
FFFE
FFFE
FFFE
FFFF
NEW PC

NOTE Refer to Table 9-7 for pin states during $\overline{\text{RESET}}$

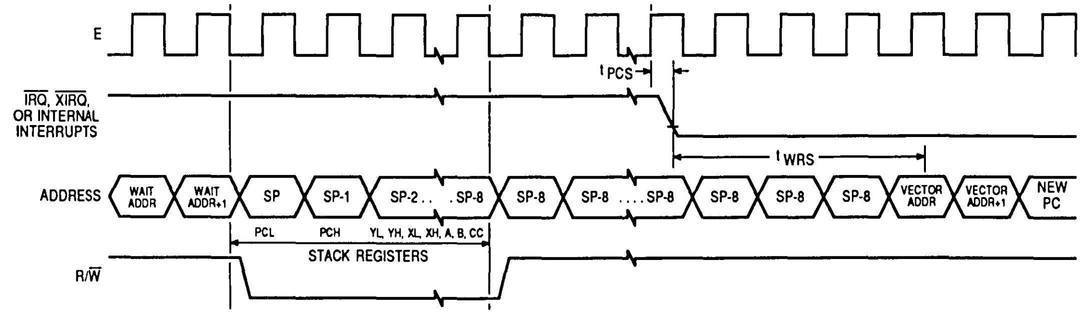
Figure 10. POR External Reset Timing Diagram


INTERNAL CLOCKS
IRQ 1
PW IRQ
IRQ, 2
XIRQ
t STOPDELAY 3
AS
E
ADDRESS 4
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
OPCODE
Resume program with instruction which follows the STOP instruction
ADDRESS 5
STOP ADDRESS
STOP ADDRESS+1
STOP ADDRESS+1
STOP ADDR+2
SP
SP-7
SP-8
SP-8
FFF2 (FFF4)
FFF3 (FFF5)
NEW PC
IRQ VECTOR
(XIRQ VECTOR)

NOTES:

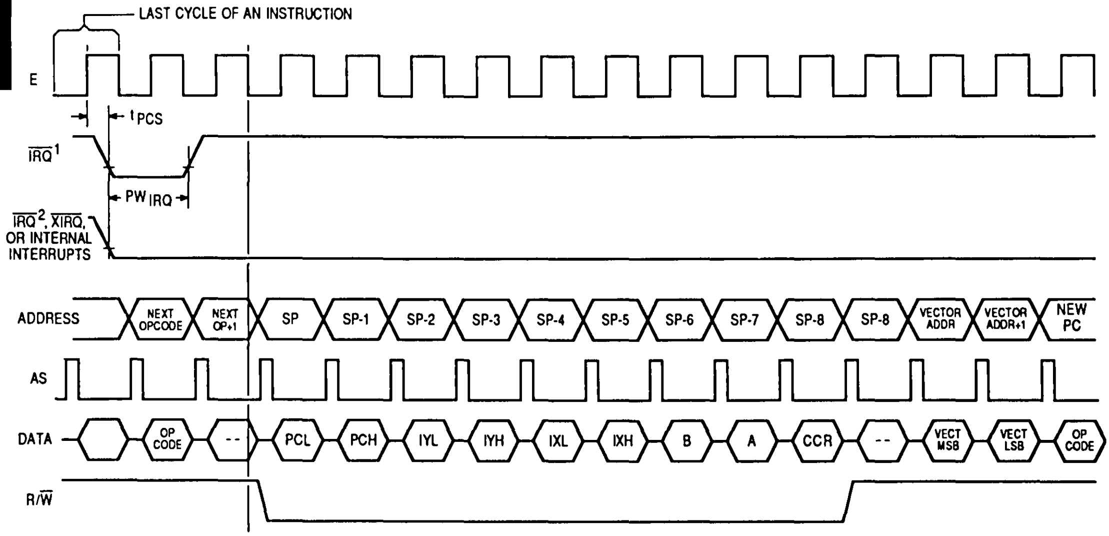
1. Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY}$ = 4064 t_{cyc} if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{\text{XIQ}}$ with X bit in CCR = 1.
5. $\overline{\text{IRQ}}$ or ($\overline{\text{XIRQ}}$ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram


E
IRQ, XIRQ, OR INTERNAL INTERRUPTS
tPCS
tWRS
ADDRESS
WAIT ADDR
WAIT ADDR+1
SP
SP-1
SP-2 . . . SP-8
SP-8
SP-8 . . . . SP-8
SP-8
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
PCL
PCH
YL, YH, XL, XH, A, B, CC
STACK REGISTERS
R/W

NOTES
1 Refer to Table 9-7 for pin states during WAIT
2 RESET will also cause recovery from WAIT

Figure 12. WAIT Recovery from Interrupt Timing Diagram


LAST CYCLE OF AN INSTRUCTION
E
tPCS
IRQ1
PWIRQ
IRQ2, XIRQ, OR INTERNAL INTERRUPTS
ADDRESS
NEXT OPCODE
NEXT OP+1
SP
SP-1
SP-2
SP-3
SP-4
SP-5
SP-6
SP-7
SP-8
SP-8
VECTOR ADDR
VECTOR ADDR+1
NEW PC
AS
DATA
OP CODE
- -
PCL
PCH
IYL
IYH
IXL
IXH
B
A
CCR
- -
VECT MSB
VECT LSB
OP CODE
R/W

NOTES
1 Edge sensitive IRQ pin (IRQE bit = 1)
2 Level sensitive IRQ pin (IRQE bit = 0)

Figure 13. Interrupt Timing Diagram

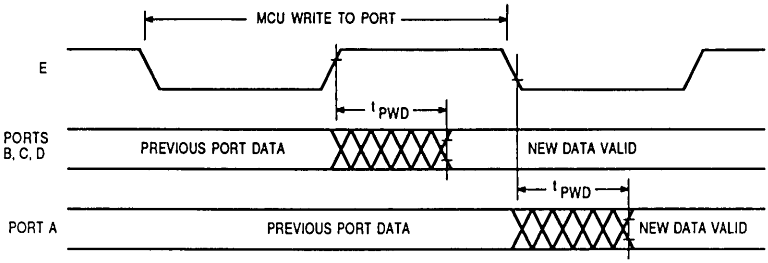
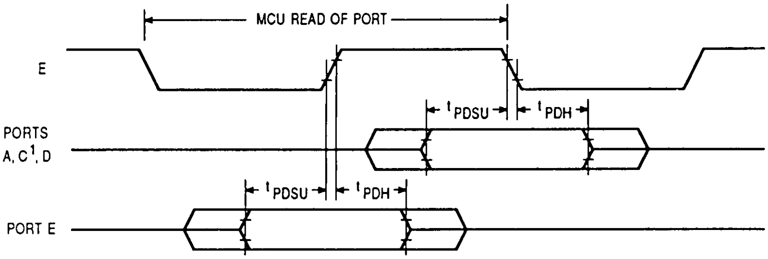

MCU WRITE TO PORT
E
t_{PWD}
PORTS B, C, D
PREVIOUS PORT DATA
NEW DATA VALID
t_{PWD}
PORT A
PREVIOUS PORT DATA
NEW DATA VALID

Figure 14. Port Write Timing Diagram


MCU READ OF PORT
E
t_{PDSU}
t_{PDH}
PORTS A, C1, D
t_{PDSU}
t_{PDH}
PORT E

NOTE1 For non-latched operation of Port C

Figure 15. Port Read Timing Diagram

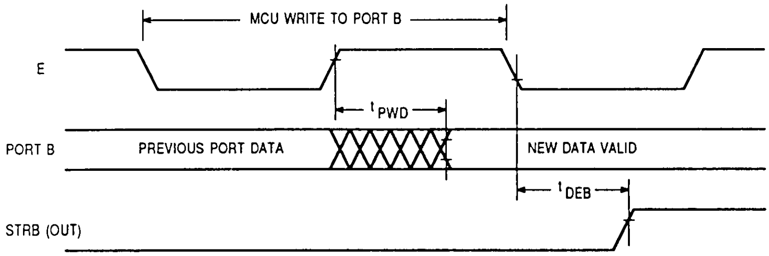

MCU WRITE TO PORT B
E
t_{PWD}
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
t_{DEB}
STRB (OUT)

Figure 16. Simple Output Strobe Timing Diagram

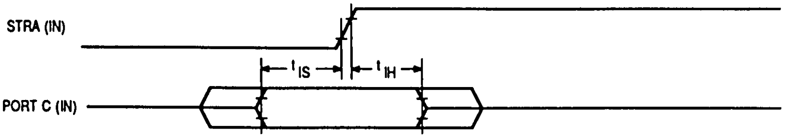
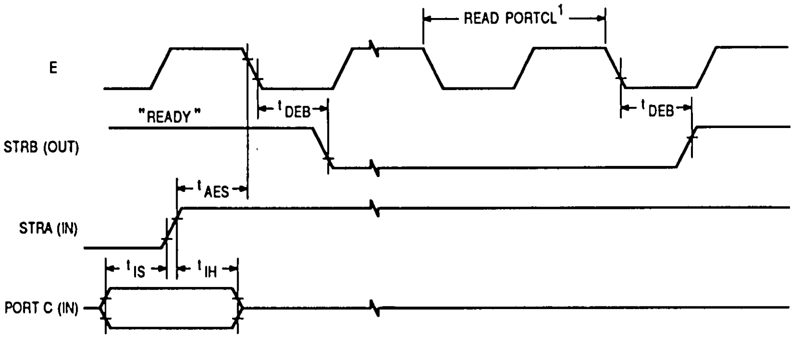
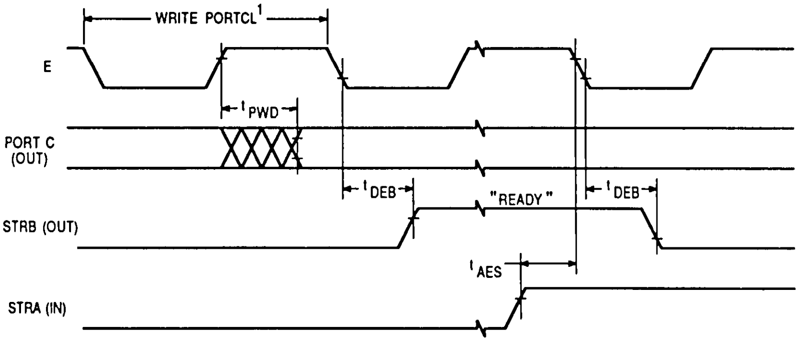

STRA (IN)
t IS
t IH
PORT C (IN)

Figure 17. Simple Input Strobe Timing Diagram


READ PORTCL¹
E
t DEB
t DEB
"READY"
STRB (OUT)
t AES
STRA (IN)
t IS
t IH
PORT C (IN)

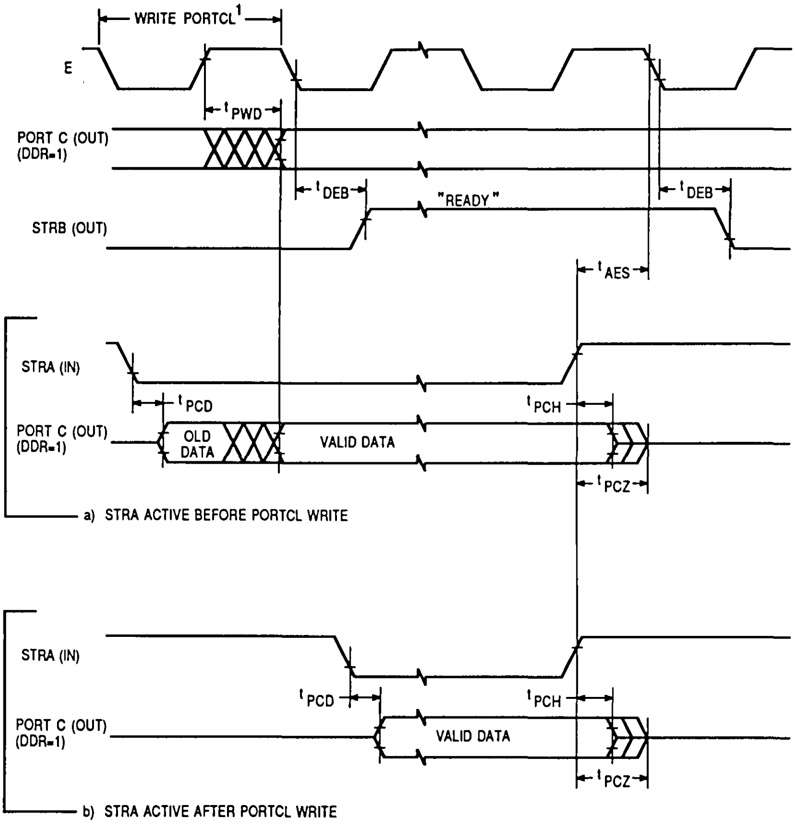
NOTES:
1 After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 18. Port C Input Handshake Timing Diagram


WRITE PORTCL¹
E
t PWD
PORT C (OUT)
t DEB
"READY"
t DEB
STRB (OUT)
t AES
STRA (IN)

NOTES
1. After reading PIOC with STAF set
2 Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 19. Port C Output Handshake Timing Diagram


WRITE PORTCL¹
E
tPWD
PORT C (OUT) (DDR=1)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
OLD DATA
VALID DATA
tPCZ
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
VALID DATA
tPCZ
b) STRA ACTIVE AFTER PORTCL WRITE

NOTES
1 After reading PIOC with STAF set
2 Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1)

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

PERIPHERAL PORT TIMING ($V_{DD}=5.0$ Vdc ± 10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 16, 18, and 19)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D $t_{PWD}=1/4\ t_{cyc}+90$ ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 17 and 18)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 17 and 18)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB $t_{DEB}=1/4\ t_{cyc}+100$ ns (See Figure 16, 18, 19, and 20)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 18, 19, 20)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 20)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 20)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 20)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:
1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3 All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	V_{DD} + 0 1	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	V_{SS} − 0.1	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion. a. E Clock b. Internal RC Oscillator	 — —	 32 —	 — t_{cyc} + 32	 t_{cyc} μs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time. a. E Clock b. Internal RC Oscillator	 — —	 12 —	 — 12	 t_{cyc} μs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7 V_{RL}, V_{RH}	— —	— —	400 1.0	nA μA

NOTES:

1 Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.

2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns$)$ see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns$)$ see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES:

1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of $1/8\ t_{cyc}$ in the above formulas where applicable:
   (a) $(1\text{-DC}) \times 1/4\ t_{cyc}$
   (b) $\text{DC} \times 1/4\ t_{cyc}$
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 22)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid (After Enable Edge)**	tv(s)	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	tho	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{rm} trs	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{fm} tfs	 — —	 100 2.0	 ns µs

*Signal production depends on software.

**Assumes 200 pF load on all SPI pins.

NOTE:

1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted

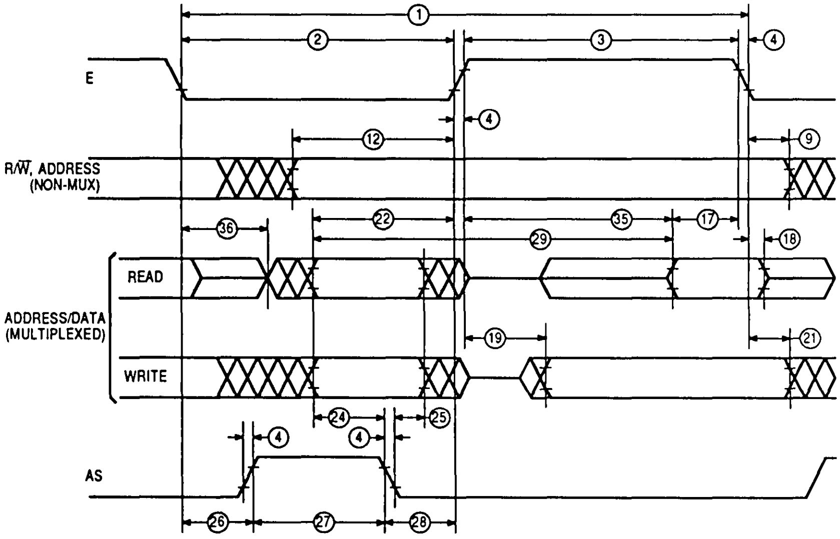
EEPROM CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic		Temperature Range			Unit
		−40 to 85°C	−40 to 105°C	−40 to 125°C	
Programming Time (see Note 1)	Under 1.0 MHz with RC Oscillator Enabled 1.0 to 2.0 MHz with RC Oscillator Disabled 2.0 MHz (or Anytime RC Oscillator Enabled)	10 20 10	15 Must Use RC 15	20 Must Use RC 20	ms
Erase Time (see Note 1)	Byte, Row, and Bulk	10	10	10	ms
Write/Erase Endurance (see Note 2)		10,000	10,000	10,000	Cycles
Data Retention (see Note 2)		10	10	10	Years

NOTES:

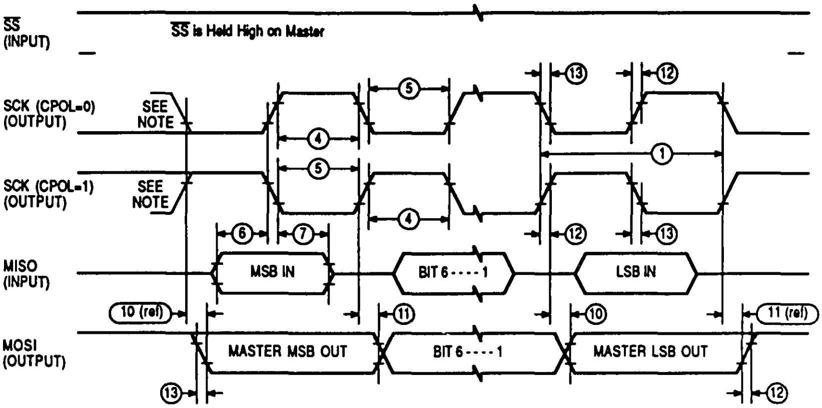
1 The RC oscillator must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1 0 MHz.

2 See current quarterly Reliability Monitor report for current failure rate information.



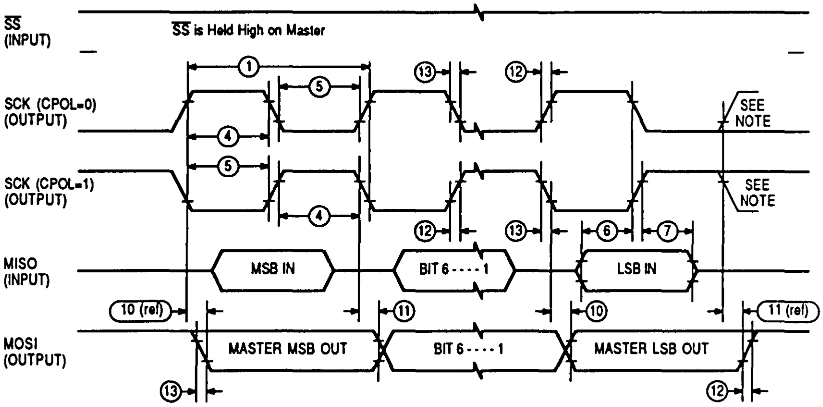
NOTE: Measurement points shown are 20% and 70% V_{DD}

Figure 21. Expansion Bus Timing Diagram


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
10 (ref)
11 (ref)
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

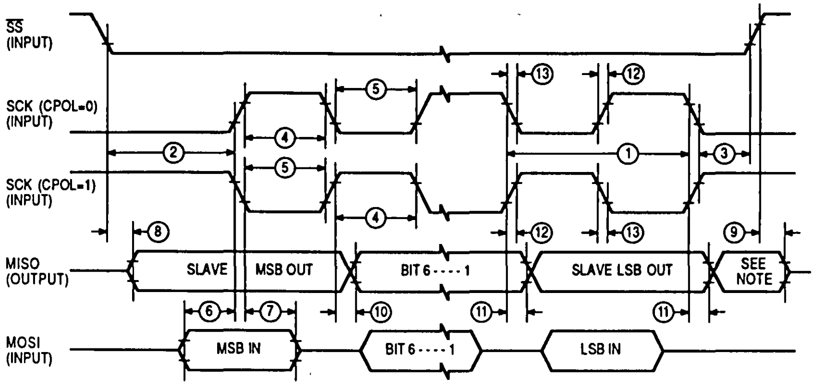
a) SPI MASTER TIMING (CPHA = 0)


SS (INPUT)
SS is Held High on Master
SCK (CPOL=0) (OUTPUT)
SEE NOTE
SCK (CPOL=1) (OUTPUT)
SEE NOTE
MISO (INPUT)
MSB IN
BIT 6 - - - - 1
LSB IN
10 (ref)
11 (ref)
MOSI (OUTPUT)
MASTER MSB OUT
BIT 6 - - - - 1
MASTER LSB OUT
①
④
⑤
⑥
⑦
⑩
⑪
⑫
⑬

NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

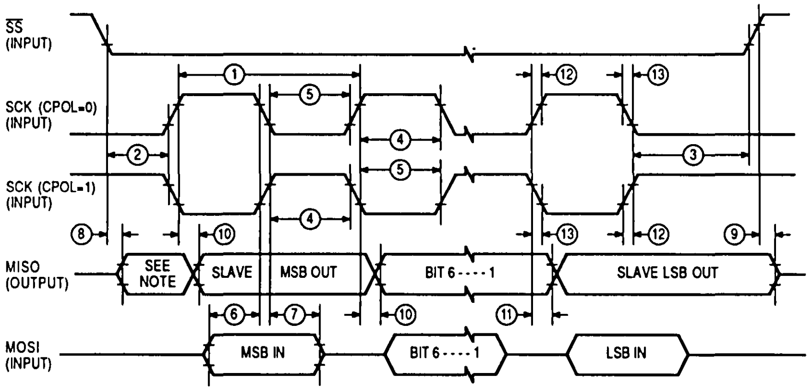
b) SPI MASTER TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
SEE NOTE
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE. Not defined but normally MSB of character just received

c) SPI SLAVE TIMING (CPHA = 0)


SS (INPUT)
SCK (CPOL=0) (INPUT)
SCK (CPOL=1) (INPUT)
MISO (OUTPUT)
MOSI (INPUT)
SEE NOTE
SLAVE
MSB OUT
BIT 6 - - - - 1
SLAVE LSB OUT
MSB IN
BIT 6 - - - - 1
LSB IN
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬

NOTE. Not defined but normally LSB of character previously transmitted

d) SPI SLAVE TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MS-DOS/PC-DOS disk file (360K)
EPROM(s): three 2532/2732 or two 2764

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field-service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MS-DOS®/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

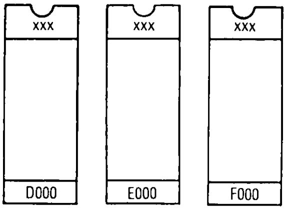
In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5 1/4-inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M68HC11 cross assemblers and linkers on IBM PC-style machines.

EPROMs

Three 2532/2732 or two 2764 type EPROM(s), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure 22 illustrates the markings for the three 2532/2732 EPROMs required to contain the customer's program.

All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Figure 23. EPROM Marking

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program ***customer supplied*** blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum-order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

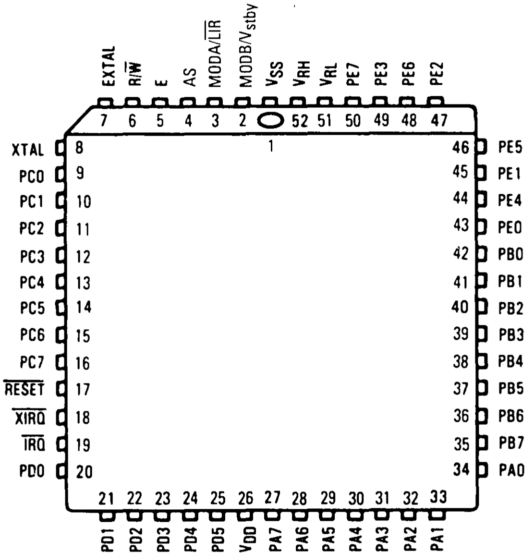
The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC811E2 HCMOS single-chip microcontroller devices.

Package Type	Temperature	CONFIG	MC Part Number
PLCC	−40° to +85°C	$FF	MC68HC811E2FN
(FN Suffix)	−40 to +105°C	$FF	MC68HC811E2VFN
	−40 to +125°C	$FF	MC68HC811E2MFN

MS-®DOS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

52-Lead Quad Package


EXTAL
R/$\overline{W}$
E
AS
MODA/$\overline{LIR}$
MODB/V_{stby}
V_{SS}
V_{RH}
V_{RL}
PE7
PE3
PE6
PE2
7
6
5
4
3
2
1
52
51
50
49
48
47
XTAL
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
$\overline{RESET}$
$\overline{XIRQ}$
$\overline{IRQ}$
PD0
8
9
10
11
12
13
14
15
16
17
18
19
20
46
45
44
43
42
41
40
39
38
37
36
35
34
PE5
PE1
PE4
PE0
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PA0
21
22
23
24
25
26
27
28
29
30
31
32
33
PD1
PD2
PD3
PD4
PD5
V_{DD}
PA7
PA6
PA5
PA4
PA3
PA2
PA1

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146818

Advance Information

REAL-TIME CLOCK PLUS RAM (RTC)

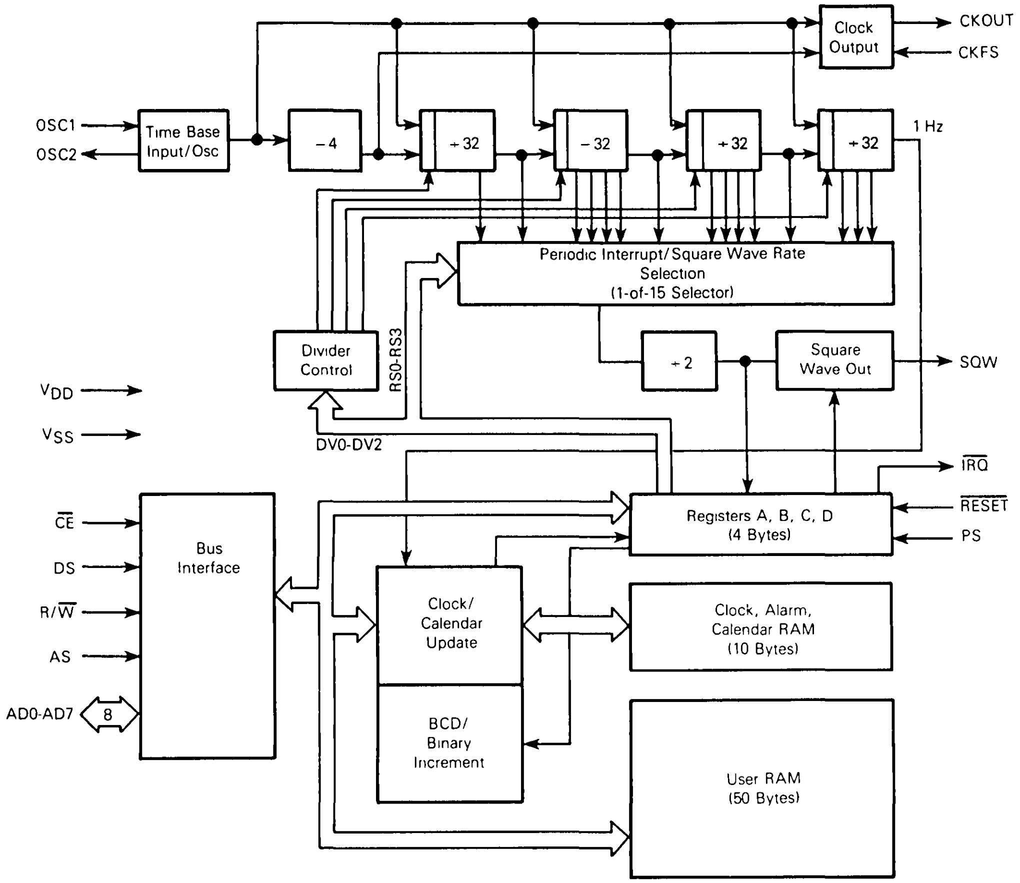
The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 μW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5 μs to 500 ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input
  - At Time Base Frequency ÷ 1 or ÷ 4
- 24-Pin Dual-In-Line Package

This document contains information on a new product Specifications and information herein are subject to change without notice

FIGURE 1 — BLOCK DIAGRAM


Clock Output
CKOUT
CKFS
OSC1
OSC2
Time Base Input/Osc
÷4
÷32
÷32
÷32
÷32
1 Hz
Periodic Interrupt/Square Wave Rate Selection (1-of-15 Selector)
Divider Control
RS0-RS3
÷2
Square Wave Out
SQW
VDD
VSS
DV0-DV2
IRQ
RESET
PS
Registers A, B, C, D (4 Bytes)
CE
DS
R/W
AS
AD0-AD7
8
Bus Interface
Clock/ Calendar Update
BCD/ Binary Increment
Clock, Alarm, Calendar RAM (10 Bytes)
User RAM (50 Bytes)

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146818 MC146818C (V_{DD} = 3.0 to 5.5 V operation)	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic	θ_{JA}	120	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32.768	32.768	kHz
Output Voltage		V_{OL}	–	0.1	V
I_{Load} < 10 µA		V_{OH}	$V_{DD}-0.1$	–	
I_{DD} – Bus Idle CKOUT = f_{osc}, C_L = 15 pF, SQW Disabled, $\overline{CE}$ = $V_{DD}-0.2$, C_L (OSC2) = 10 pF f_{osc} = 32.768 kHz		I_{DD3}	–	50	µA
I_{DD} – Quiescent f_{osc} = DC, OSC1 = DC, All Other Inputs = $V_{DD}-0.2$ V, No Clock		I_{DD4}	–	50	µA
Output High Voltage (I_{Load} = −0.25 mA, All Outputs)		V_{OH}	2.7	–	V
Output Low Voltage (I_{Load} = 0.25 mA, All Outputs)		V_{OL}	–	0.3	V
Input High Voltage	AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, $\overline{RESET}$, CKFS, PS, OSC1	V_{IH}	2.1 2.5	V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)		V_{IL}	V_{SS}	0.5	V
Input Current	All Inputs	I_{in}	–	±1	µA
Three-State Leakage	$\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	µA

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32.768	4194.304	kHz
Output Voltage		V_{OL}	–	0.1	V
I_{Load} < 10 µA		V_{OH}	$V_{DD}-0.1$	–	
I_{DD} – Bus Idle (External Clock) CKOUT = f_{osc}, C_L = 15 pF, SQW Disabled, $\overline{CE}$ = $V_{DD}-0.2$, C_L (OSC2) = 10 pF f_{osc} = 4.194304 MHz f_{osc} = 1.048516 MHz f_{osc} = 32.768 kHz		 I_{DD1} I_{DD2} I_{DD3}	 – – –	 3 800 50	 mA µA µA
I_{DD} – Quiescent f_{osc} = DC, OSC1 = DC, All Other Inputs = $V_{DD}-0.2$ V, No Clock		I_{DD4}	–	50	µA
Output High Voltage (I_{Load} = −1.6 mA, AD0-AD7, CKOUT) (I_{Load} = −1.0 mA, SQW)		V_{OH}	4.1	–	V
Output Low Voltage (I_{Load} = 1.6 mA, AD0-AD7, CKOUT) (I_{Load} = 1.0 mA, $\overline{IRQ}$ and SQW)		V_{OL}	–	0.4	V
Input High Voltage	CKFS, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, PS $\overline{RESET}$ OSC1	V_{IH}	$V_{DD}-2.0$ $V_{DD}-0.8$ $V_{DD}-1.0$	V_{DD} V_{DD} V_{DD}	V
Input Low Voltage	AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$ CKFS, PS, $\overline{RESET}$ OSC1	V_{IL}	V_{SS} V_{SS} V_{SS}	0.8 0.8 0.8	V
Input Current	All Inputs	I_{in}	–	±1	µA
Three-State Leakage	$\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	µA

BUS TIMING

Ident. Number	Characteristics	Symbol	$V_{DD}=3.0$ V 50 pF Load		$V_{DD}=5.0$ V ± 10% 2 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	–	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High	PW_{EL}	1000	–	300	–	ns
3	Pulse Width, DS/E High or $\overline{RD}$/$\overline{WR}$ Low	PW_{EH}	1500	–	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	10	–	ns
13	R/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	–	80	–	ns
14	Chip Enable Setup Time Before AS/ALE Fall	t_{CS}	200	*	55	*	ns
15	Chip Enable Hold Time	t_{CH}	10	–	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	–	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	–	50	–	ns
25	Muxed Address Hold Time	t_{AHL}	100	–	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	–	50	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	–	135	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	–	60	–	ns

FIGURE 3 — BUS READ TIMING COMPETITOR MULTIPLEXED BUS

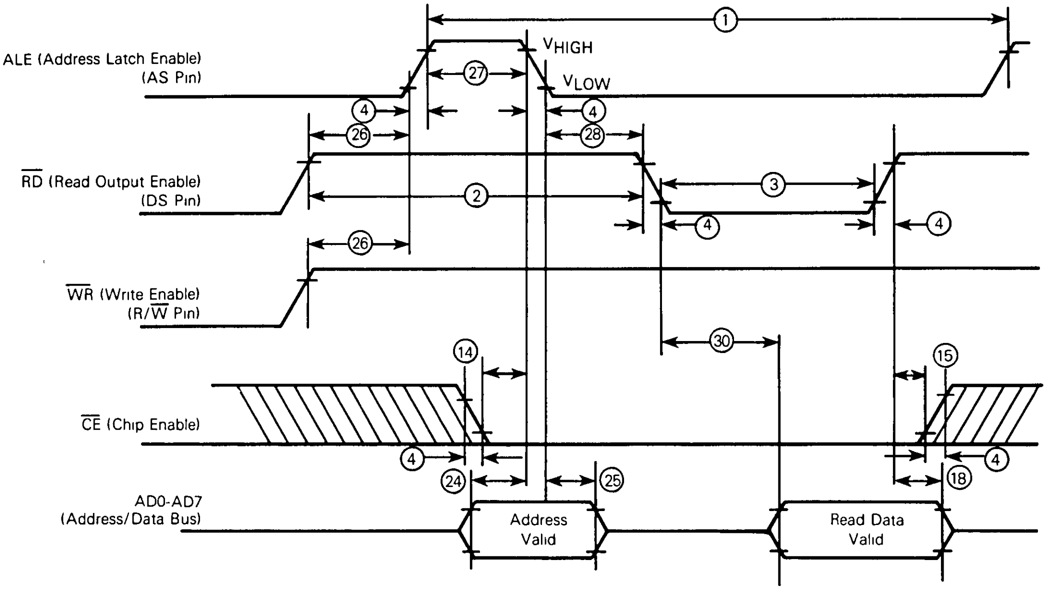
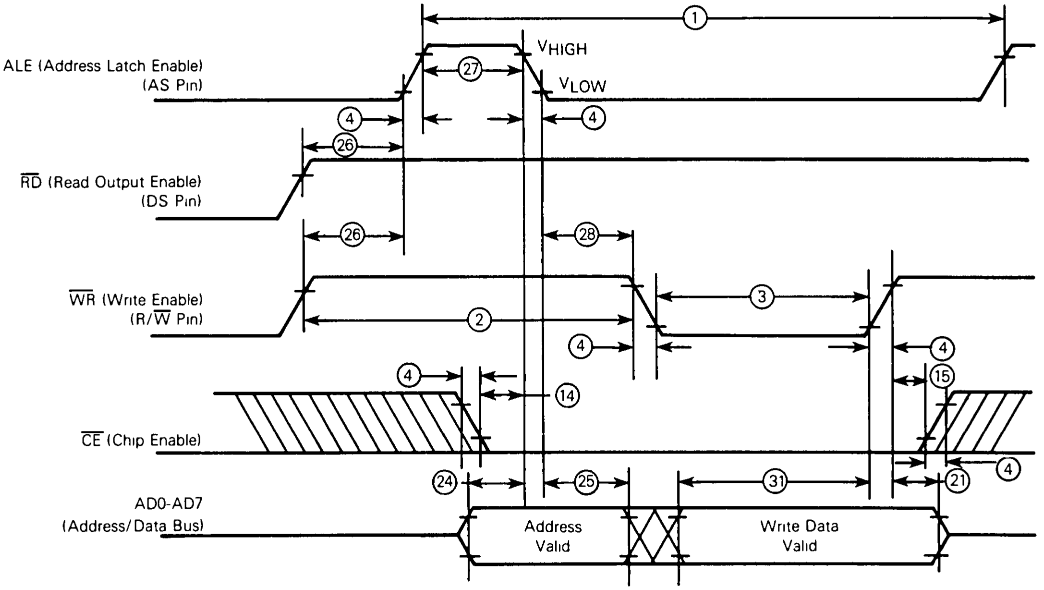

ALE (Address Latch Enable)
(AS Pin)
RD (Read Output Enable)
(DS Pin)
WR (Write Enable)
(R/W Pin)
CE (Chip Enable)
AD0-AD7
(Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑭
⑮
⑱
㉔
㉕
㉖
㉗
㉘
㉚
Address Valid
Read Data Valid

FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS

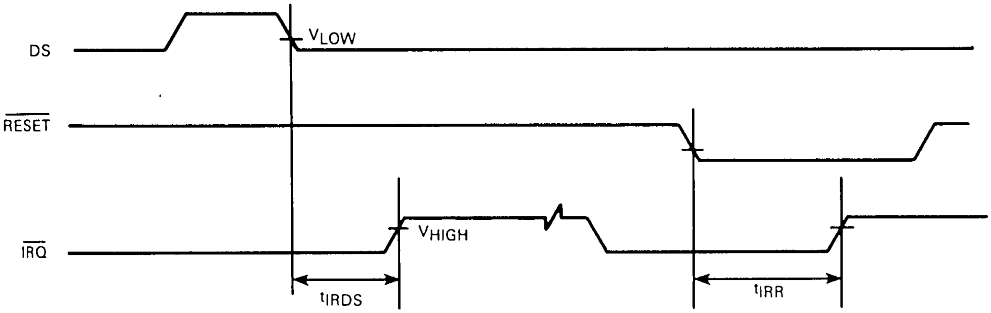

ALE (Address Latch Enable)
(AS Pin)
RD (Read Output Enable)
(DS Pin)
WR (Write Enable)
(R/W Pin)
CE (Chip Enable)
AD0-AD7
(Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑭
⑮
㉑
㉔
㉕
㉖
㉗
㉘
㉛
Address Valid
Write Data Valid

NOTE $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

TABLE 1 — SWITCHING CHARACTERISTICS (V_{DD} = 5 0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

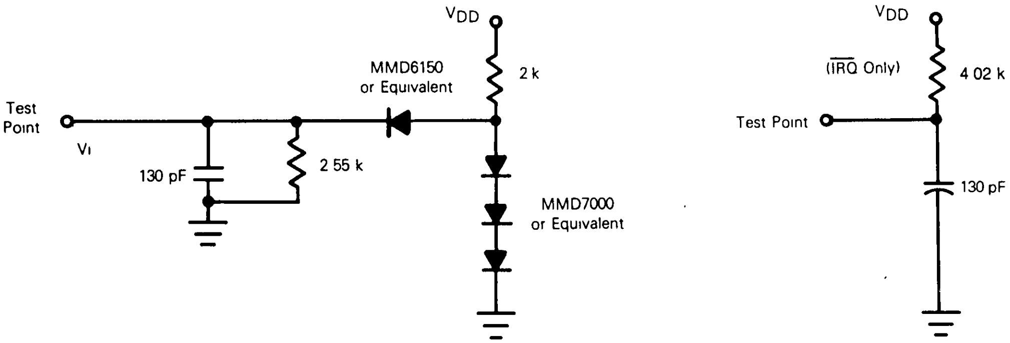
Description	Symbol	Min	Max	Unit
Oscillator Startup	t_{RC}	—	100	ms
Reset Pulse Width	t_{RWL}	5	—	μs
Reset Delay Time	t_{RLH}	5	—	μs
Power Sense Pulse Width	t_{PWL}	5	—	μs
Power Sense Delay Time	t_{PLH}	5	—	μs
$\overline{IRQ}$ Release from DS	t_{IRDS}	—	2	μs
$\overline{IRQ}$ Release from $\overline{RESET}$	t_{IRR}	—	2	μs
VRT Bit Delay	t_{VRTD}	—	2	μs

FIGURE 5 — $\overline{IRQ}$ RELEASE DELAY


DS
V_{LOW}
$\overline{RESET}$
$\overline{IRQ}$
V_{HIGH}
t_{IRDS}
t_{IRR}

NOTE V_{HIGH} = V_{DD} − 2 0 V, V_{LOW} = 0 8 V, for V_{DD} = 5 0 V ± 10%

FIGURE 6 — TTL EQUIVALENT TEST LOAD


V_{DD}
MMD6150
or Equivalent
2 k
Test
Point
Vı
130 pF
2 55 k
MMD7000
or Equivalent
V_{DD}
($\overline{IRQ}$ Only)
4 02 k
Test Point
130 pF

All Outputs Except OSC2 (See Figure 10)

FIGURE 7 — POWER-UP

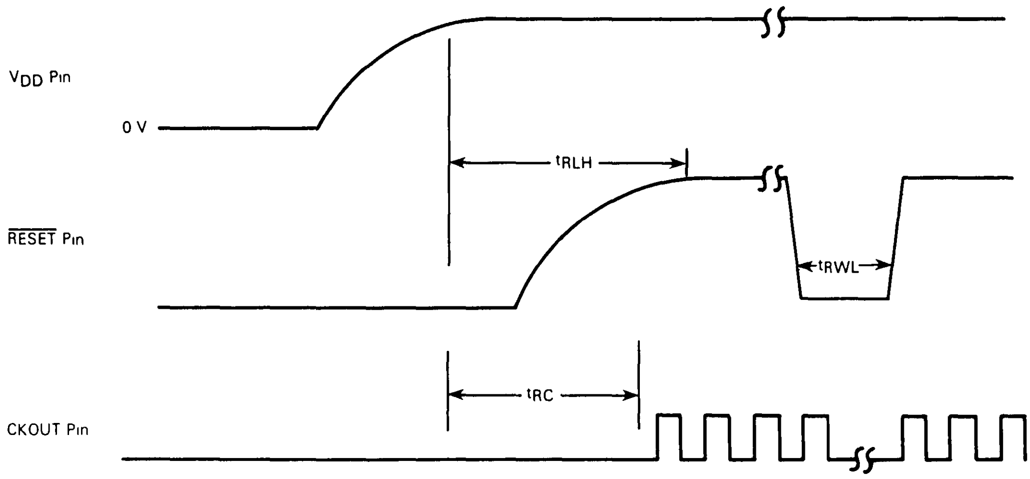
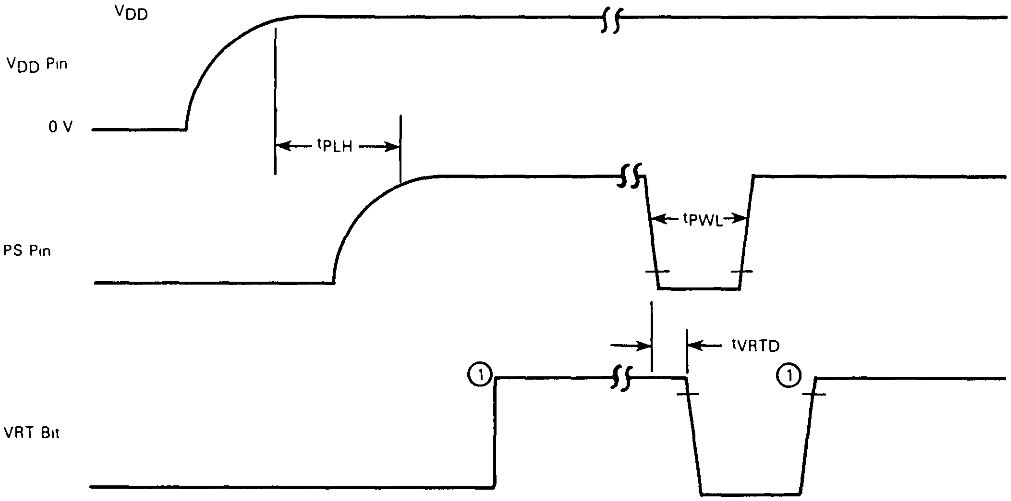

VDD Pin
0 V
tRLH
RESET Pin
tRWL
tRC
CKOUT Pin

FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT


VDD
VDD Pin
0 V
tPLH
PS Pin
tPWL
tVRTD
①
①
VRT Bit

① The VRT bit is set to a "1" by reading Register d The VRT bit can only be cleared by pulling the PS pin low (see **REGISTER D ($0D)**).

MOTEL

The MOTEL circuit is a new concept that permits the MC146818 to be directly interfaced with many types of microprocessors No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

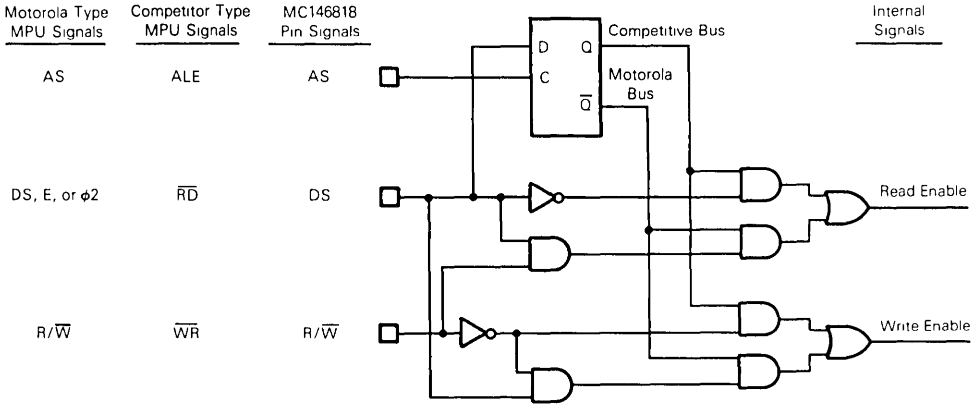
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated by the Motorola MC6800 and the other by the Intel 8080 and its companion part, the 8228.

The MOTEL circuit (for MOTorola and IntEL bus compatibility) is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry standard bus structure is now available. The MOTEL concept is shown logically in Figure 9

MOTEL selects one of two interpretations of two pins In the Motorola case, DS and R/$\overline{W}$ are gated together to produce the internal read enable The internal write enable is a similar gating of the inverse of R/$\overline{W}$ With competitor buses, the inversion of $\overline{RD}$ and $\overline{WR}$ create functionally identical internal read and write enable signals

The MC146818 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{RD}$ pin. Since DS is always low and $\overline{RD}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

FIGURE 9 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the MC146818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins, V_{DD} being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables

OSC1, OSC2 — TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 10. The internal time-base frequency to be used is chosen in Register A

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4 194304 MHz or 1 048576 MHz frequencies. The crystal connections are shown in Figure 11 and the crystal characteristics in Figure 12.

CKOUT — CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4 A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS — CLOCK OUT FREQUENCY SELECT, INPUT

When the CKFS pin is tied to V_{DD} it causes CKOUT to be the same frequency as the time base at the OSC1 pin When CKFS is tied to V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS

FIGURE 10 — EXTERNAL TIME-BASE CONNECTION

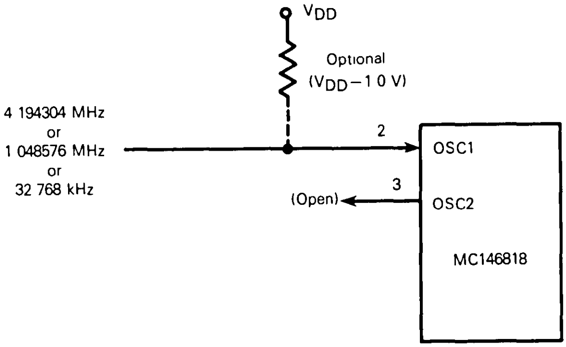
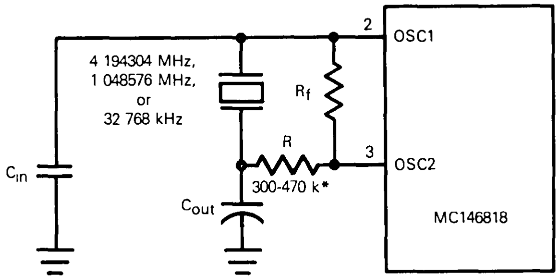

VDD
Optional
(VDD−1 0 V)
4 194304 MHz
or
1 048576 MHz
or
32 768 kHz
2
OSC1
3
(Open)
OSC2
MC146818

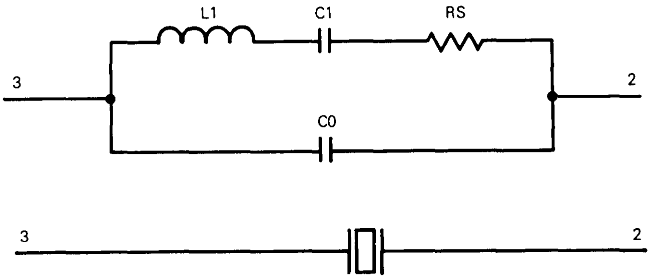
FIGURE 11 — CRYSTAL OSCILLATOR CONNECTION


2
OSC1
4 194304 MHz,
1 048576 MHz,
or
32 768 kHz
Rf
R
3
OSC2
Cin
300-470 k*
Cout
MC146818

*32 768 kHz Only — Consult Crystal Manufacturer's Specification

FIGURE 12 — CRYSTAL PARAMETERS

Crystal Equivalent Circuit


L1
C1
RS
3
2
C0
3
2

f_{osc}	4.194304 MHz	1.048576 MHz	32.768 kHz
RS (Maximum)	75 Ω	700 Ω	50 k
C0 (Maximum)	7 pF	5 pF	1 7 pF
C1	0 012 pF	0 008 pF	0 003 pF
Q	50 k	35 k	30 k
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
R	—	—	300-470 k
R_f	10 M	10 M	22 M

TABLE 2 — CLOCK OUTPUT FREQUENCIES

Time Base (OSC1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4 194304 MHz	High	4.194304 MHz
4 194304 MHz	Low	1 048576 MHz
1 048576 MHz	High	1 048576 MHz
1.048576 MHz	Low	262 144 kHz
32 768 kHz	High	32 768 kHz
32.768 kHz	Low	8 192 kHz

SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B

AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the MC146818 since the bus reversal from address to data is occurring during the internal RAM access time

The address must be valid just prior to the fall of AS/ALE at which time the MC146818 latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146818 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to the high-impedance state) when DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case

AS — MULTIPLEXED ADDRESS STROBE, INPUT

A positive going multiplexed address strobe pulse serves to demultiplex the bus The falling edge of AS or ALE causes the address to be latched within the MC146818 The automatic MOTEL circuit in the MC146818 also latches the state of the DS pin with the falling edge of AS or ALE.

DS — DATA STROBE OR READ, INPUT

The DS pin has two interpretations via the MOTEL circuit. When emanating from a Motorola type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from the competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the MC146818, latches the state of the DS pin on the falling edge of AS/ALE. When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus processors To ensure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

R/$\overline{W}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{W}$ pin in one of two ways When a Motorola type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ during DS

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor type processors The MOTEL circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write (W) pulse on many generic RAMs

$\overline{CE}$ — CHIP ENABLE, INPUT

The chip-enable ($\overline{CE}$) signal must be asserted (low) for a bus cycle in which the MC146818 is to be accessed $\overline{CE}$ is not latched and must be stable during DS and AS (Motorola case of MOTEL) and during $\overline{RD}$ and $\overline{WR}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the MC146818 When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state

When $\overline{CE}$ is high, all address, data, DS, and R/$\overline{W}$ inputs from the processor are disconnected within the MC146818 This permits the MC146818 to be isolated from a powered-down processor. When $\overline{CE}$ is held high, an unpowered device cannot receive power through the input pins from the real-time clock power source Battery power consumption can thus be reduced by using a pullup resistor or active clamp on $\overline{CE}$ when the main power is off When $\overline{CE}$ is not used, it should be grounded

$\overline{IRQ}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{IRQ}$ pin is an active low output of the MC146818 that may be used as an interrupt input to a processor The $\overline{IRQ}$ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set To clear the $\overline{IRQ}$ pin, the processor program normally reads Register C The $\overline{RESET}$ pin also clears pending interrupts

When no interrupt conditions are present, the $\overline{IRQ}$ level is in the high-impedance state Multiple interrupting devices may thus be connected to an $\overline{IRQ}$ bus with one pullup at the processor.

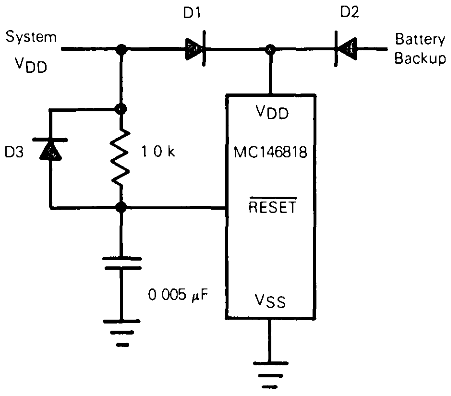
$\overline{RESET}$ — RESET, INPUT

The $\overline{RESET}$ pin does not affect the clock, calendar, or RAM functions On powerup, the $\overline{RESET}$ pin must be held low for the specified time, t_{RLH}, in order to allow the power supply to stabilize. Figure 13 shows a typical representation of the $\overline{RESET}$ pin circuit

When $\overline{RESET}$ is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Update ended Interrupt Enable (UIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible

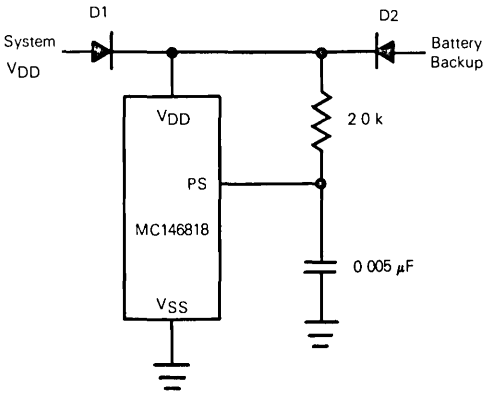
FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR $\overline{\text{RESET}}$



D1 = MBD701 (Schottky) or Equivalent
D2 = D3 = 1N4148 or Equivalent

Note If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements

FIGURE 14 — TYPICAL POWERUP DELAY CIRCUIT FOR POWER SENSE



D1 = MBD701 (Schottky) or Equivalent
D2 = 1N4148 or Equivalent

g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{\text{IRQ}}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to zero.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D When the PS pin is low the VRT bit is cleared to zero

When using the VRT feature during powerup, the PS pin must be externally held low for the specified t_{PLH} time As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed PS must go high after powerup to allow the VRT bit to be set by a read of register D

POWER-DOWN CONSIDERATIONS

In most systems, the MC146818 must continue to keep time when system power is removed In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability

The chip enable ($\overline{\text{CE}}$) pin controls all bus inputs (R/$\overline{\text{W}}$, DS, AS, AD0-AD7) $\overline{\text{CE}}$, when negated, disallows any unintended modification of the RTC data by the bus $\overline{\text{CE}}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part

ADDRESS MAP

Figure 15 shows the address map of the MC146818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except for the following: 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD)

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23 The 24/12 bit cannot be changed without reinitializing the hour locations When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition If any of the 10 bytes are read at this time, the data outputs are undefined The update lockout time is 248 μs at the 4.194304 MHz and 1 048567 MHz time bases and 1948 μs for the 32 768 kHz time base The Update Cycle section shows how to accommodate the update cycle in the processor program

FIGURE 15 — ADDRESS MAP

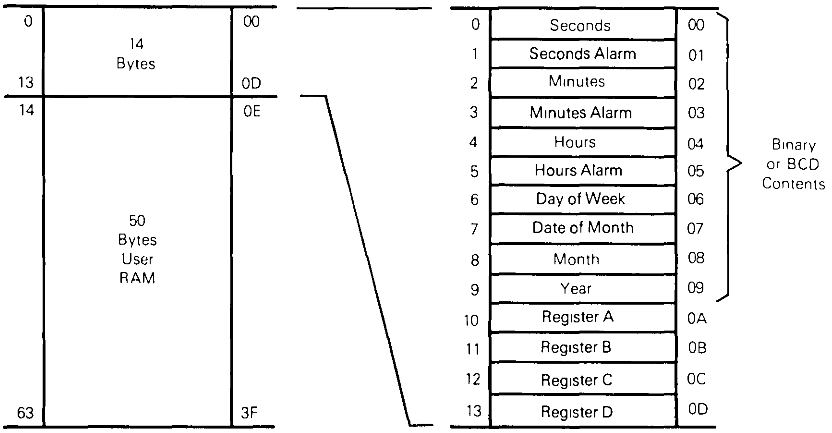

0
13
14
63
14 Bytes
50 Bytes User RAM
00
0D
0E
3F
0 Seconds 00
1 Seconds Alarm 01
2 Minutes 02
3 Minutes Alarm 03
4 Hours 04
5 Hours Alarm 05
6 Day of Week 06
7 Date of Month 07
8 Month 08
9 Year 09
10 Register A 0A
11 Register B 0B
12 Register C 0C
13 Register D 0D
Binary or BCD Contents

TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

Address Location	Function	Decimal Range	Range		Example*	
			Binary Data Mode	BCD Data Mode	Binary Data Mode	BCD Data Mode
0	Seconds	0-59	$00-$3B	$00-$59	15	21
1	Seconds Alarm	0-59	$00-$3B	$00-$59	15	21
2	Minutes	0-59	$00-$3B	$00-$59	3A	58
3	Minutes Alarm	0-59	$00-$3B	$00-$59	3A	58
4	Hours (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	$00-$17	$00-$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	$00-$17	$00-23	05	05
6	Day of the Week Sunday = 1	1-7	$01-$07	$01-$07	05	05
7	Date of the Month	1-31	$01-$1F	$01-$31	0F	15
8	Month	1-12	$01-$0C	$01-$12	02	02
9	Year	0-99	$00-$63	$00-$99	4F	79

*Example 5 58.21 Thursday 15 February 1979 (time is AM)

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes The "don't care" codes in all three alarm bytes create an interrupt every second

STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the MC146818 They can be used by the processor program, and are fully available during the update cycle

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program

When further CMOS RAM is needed, additional MC146818s may be included in the system The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur With the dividers clear, the available user RAM is extended to 59 bytes The high-order bit of the seconds byte, bit 7 of Register A, and all bits of Registers C and D cannot effectively be used as general purpose RAM

INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day The periodic interrupt may be selected for rates from half-a-second to 30 517 μs The update-ended interrupt may be used to indicate to the program that an update cycle is completed Each of these independent interrupt conditions are described in greater detail in other sections

The processor program selects which interrupts, if any, it wishes to receive Three bits in Register B enable the three interrupts Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{\text{IRQ}}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read

However, there is one precaution The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{\text{IRQ}}$ pin is asserted low. $\overline{\text{IRQ}}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{\text{IRQ}}$ pin is being driven low

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set Again, more than one interrupt-flag bit may be set

DIVIDER STAGES

The MC146818 has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controller by three divider bus (DV2, DV1, and DV0) in Register A.

DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4 194304 MHz, 1 048576 MHz, or 32 768 kHz). The divider chain may be held reset, which allows precision setting of the time When the divider is changed from reset to an operating time base, the first update cycle is one-half second later The divider-control bits are also used to facilitate testing the MC146818

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4 194304 MHz	0	0	0	Yes	–	N = 0
1 048576 MHz	0	0	1	Yes	–	N = 2
32 768 kHz	0	1	0	Yes	–	N = 7
Any	1	1	0	No	Yes	–
Any	1	1	1	No	Yes	–

Note Other combinations of divider bits are used for test purposes only

SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5 The SQW frequency selection shares the 1-of-15 selector with periodic interrupts

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B Altering the divider, square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution The square-wave output pin has a number of potential uses For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control

PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30 517 μs The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency Changing one also changes the other But each function may be separately enabled so that a program could switch between the two features or use both The SQW pin is enabled by the SQWE bit in Register B Similarly the periodic interrupt is enabled by the PIE bit in Register B

Periodic interrupt is usable by practically all real-time systems It can be used to scan for all forms of inputs from contact closures to serial receive bits or bytes It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Select Bits Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate t_{PI}	SQW Output Frequency	Periodic Interrupt Rate t_{PI}	SQW Output Frequency
0	0	0	0	None	None	None	None
0	0	0	1	30 517 μs	32 768 kHz	3 90625 ms	256 Hz
0	0	1	0	61 035 μs	16 384 kHz	7 8125 ms	128 Hz
0	0	1	1	122 070 μs	8 192 kHz	122 070 μs	8 192 kHz
0	1	0	0	244 141 μs	4 096 kHz	244 141 μs	4 096 kHz
0	1	0	1	488 281 μs	2 048 kHz	488 281 μs	2 048 kHz
0	1	1	0	976 562 μs	1 024 kHz	976 562 μs	1 024 kHz
0	1	1	1	1 953125 ms	512 Hz	1 953125 ms	512 Hz
1	0	0	0	3 90625 ms	256 Hz	3 90625 ms	256 Hz
1	0	0	1	7 8125 ms	128 Hz	7 8125 ms	128 Hz
1	0	1	0	15 625 ms	64 Hz	15 625 ms	64 Hz
1	0	1	1	31 25 ms	32 Hz	31 25 ms	32 Hz
1	1	0	0	62 5 ms	16 Hz	62 5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

UPDATE CYCLE

The MC146818 executes an update cycle once-per-second, assuming one of the proper time bases is in place, the DV0-DV2 divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 µs while a 32.768 kHz time base update cycle takes 1984 µs. During the update cycle, the time, calendar, and alarm bytes are not accessable by the processor program The MC146818 protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle If the processor reads these RAM locations before the update is complete the output will be undefined The update in progress (UIP) status bit is set during the interval

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts Three methods of accommodating nonavailability during update are usable by the program In discussing the three methods it is assumed that at random points user programs are able to call a subroutine to obtain the time of day

The first method of avoiding the update cycle uses the update-ended interrupt If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information During this time a display could be updated or the information could be transfered to continuously available RAM Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not The UIP bit will pulse once-per-second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 µs later. Therefore, if a low is read on the UIP bit, the user has at least 244 µs before the time/calendar data will be changed If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 µs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 16) Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to ensure that data is not read during the update cycle.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

REGISTERS

The MC146818 has four registers which are accessible to the processor program The four registers are also fully accessible during the update cycle

REGISTER A ($0A)

MSB — LSB

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

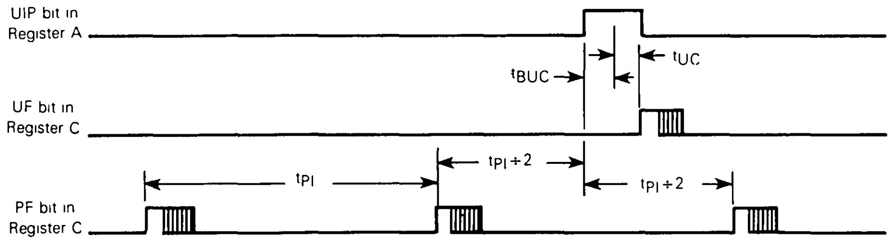
Read/Write Register except UIP

UIP — The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin When UIP is a "0" the update cycle is not in progress and will not be for at least 244 µs (for all time bases) This is detailed in Table 6 The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibit any update cycle and then clear the UIP status bit.

TABLE 6 — UPDATE CYCLE TIMES

UIP Bit	Time Base (OSC1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4 194304 MHz	248 µs	–
1	1 048576 MHz	248 µs	–
1	32.768 kHz	1984 µs	–
0	4.194304 MHz	–	244 µs
0	1 048576 MHz	–	244 µs
0	32.768 kHz	–	244 µs

FIGURE 16 — UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIPS



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62 5 ms, etc per Table 5)
t_{UC} = Update Cycle Time (248 µs or 1984 µs)
t_{BUC} = Delay Time Before Update Cycle (244 µs)

DV2, DV1, DV0 — Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

RS3, RS2, RS1, RS0 — The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate,

quency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

DM — The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12 — The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

DSE — The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two

REGISTER D ($0D)

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read Only
VRT	0	0	0	0	0	0	0	Register

VRT – The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read only bit which is not modified by the RESET pin The VRT bit can only be set by reading Register D.

b6 TO b0 – The remaining bits of Register D are unused They cannot be written, but are always read as "0's."

TYPICAL INTERFACING

The MC146818 is best suited for use with microprocessors which generate an address-then-data multiplexed bus Figures 17 and 18 show typical interfaces to bus-compatible processors These interfaces assume that the address decoding can be done quickly However, if standard metal-gate CMOS gates are used the CE setup time may be violated. Figure 19 illustrates an alternative method of chip selection which will accommodate such slower decoding.

The MC146818 can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 20. Non-multiplexed bus microprocessors can be interfaced with additional support

There is one method of using the multiplexed bus MC146818 with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together

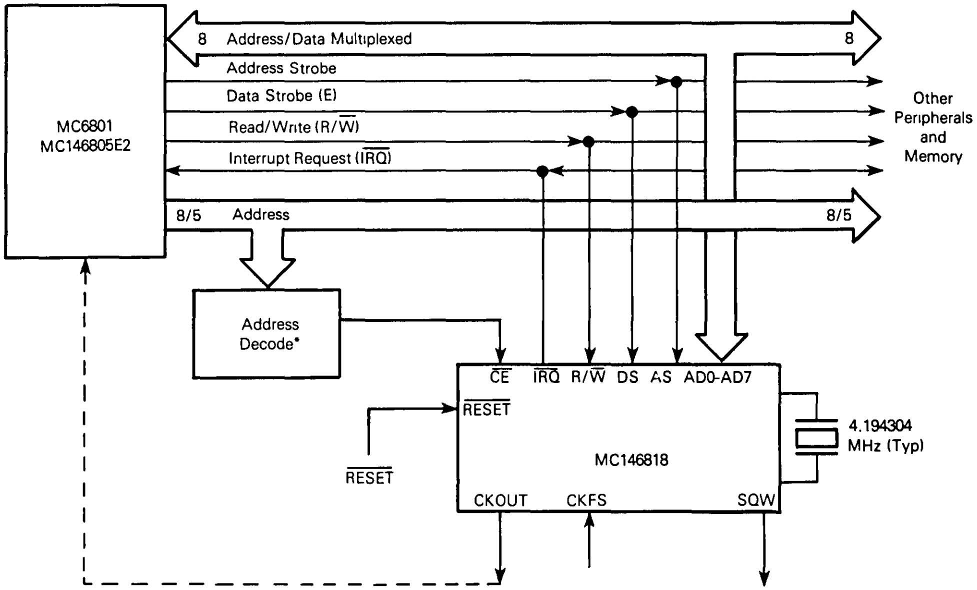
An example using either the Motorola MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 21

Figure 22 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data

Accumulator A. The address of the RTC to be accessed
Accumulator B Write: The data to be written.
Read. The data read from the RTC.

The RTC is mapped to two consecutive memory locations – RTC and RTC + 1 as shown in Figure 21.

FIGURE 17 – MC146818 INTERFACED WITH MOTOROLA COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS


MC6801
MC146805E2
8
Address/Data Multiplexed
8
Address Strobe
Data Strobe (E)
Read/Write (R/W)
Interrupt Request (IRQ)
8/5
Address
8/5
Other Peripherals and Memory
Address Decode*
CE
IRQ
R/W
DS
AS
AD0-AD7
RESET
MC146818
RESET
CKOUT
CKFS
SQW
4.194304 MHz (Typ)

*High-Speed Silicon-Gate CMOS or TTL Address Decoding

FIGURE 18 — MC146818 INTERFACED WITH COMPETITOR COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

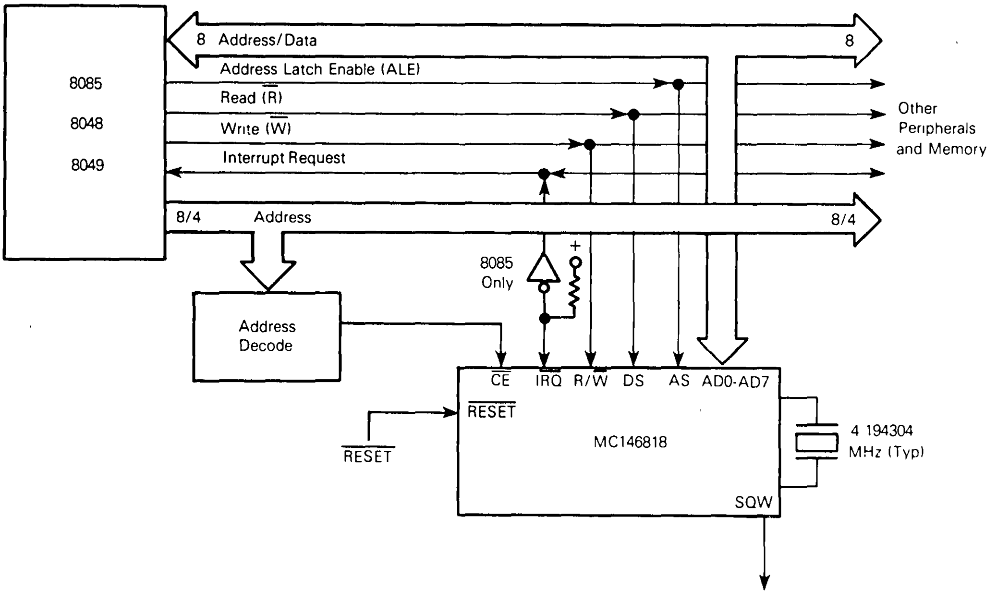
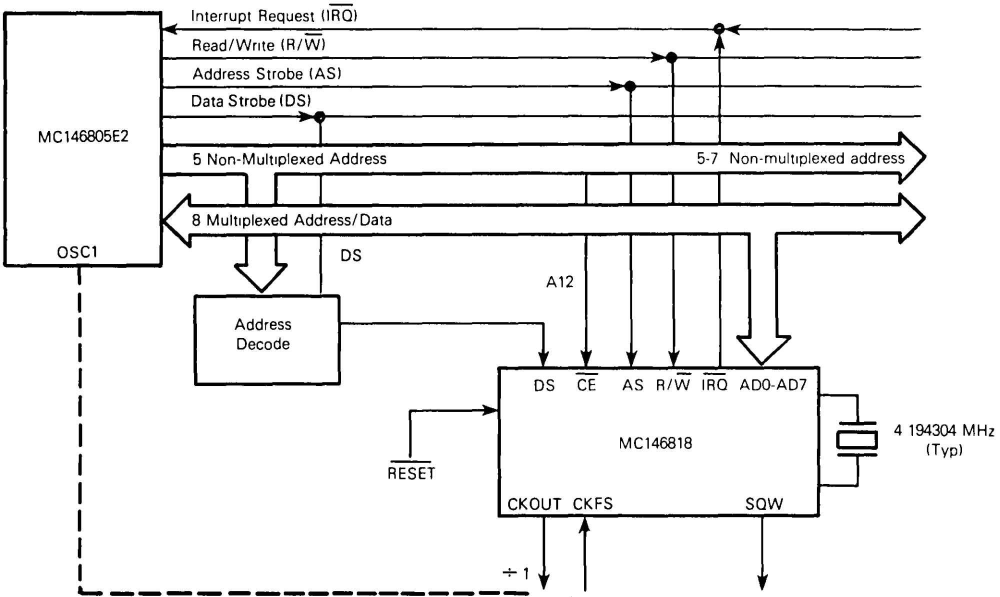

8085
8048
8049
8 Address/Data
8
Address Latch Enable (ALE)
Read ($\overline{R}$)
Write ($\overline{W}$)
Interrupt Request
8/4
Address
8/4
Other Peripherals and Memory
8085 Only
+
Address Decode
$\overline{CE}$
$\overline{IRQ}$
R/$\overline{W}$
DS
AS
AD0-AD7
$\overline{RESET}$
MC146818
SQW
$\overline{RESET}$
4 194304 MHz (Typ)

FIGURE 19 — MC146818 INTERFACE WITH MC146805E2 CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING


MC146805E2
OSC1
Interrupt Request ($\overline{IRQ}$)
Read/Write (R/$\overline{W}$)
Address Strobe (AS)
Data Strobe (DS)
5 Non-Multiplexed Address
5-7 Non-multiplexed address
8 Multiplexed Address/Data
DS
A12
Address Decode
DS
$\overline{CE}$
AS
R/$\overline{W}$
$\overline{IRQ}$
AD0-AD7
MC146818
CKOUT
CKFS
SQW
$\overline{RESET}$
4 194304 MHz (Typ)
÷ 1
V_{DD}

This illustrates the use of CMOS gating for address decoding

FIGURE 20 — MC146818 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER

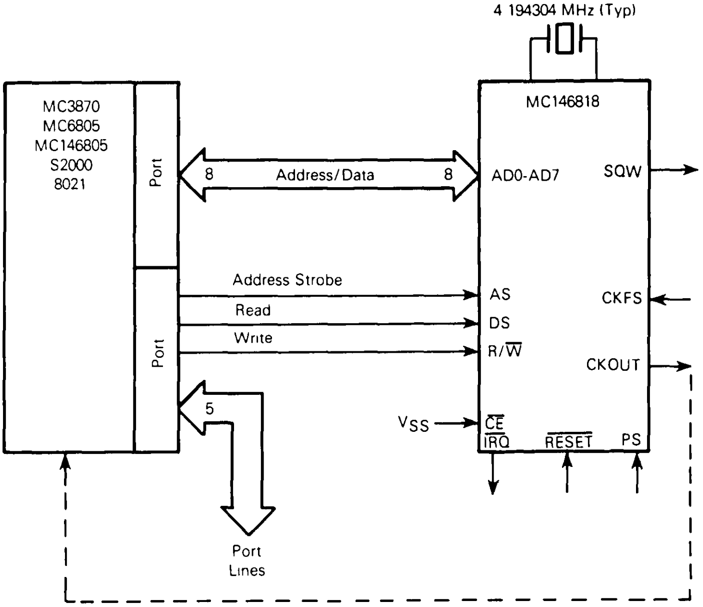

4 194304 MHz (Typ)
MC3870
MC6805
MC146805
S2000
8021
Port
Port
8
Address/Data
8
Address Strobe
Read
Write
5
Port Lines
MC146818
AD0-AD7
SQW
AS
DS
R/$\overline{W}$
CKFS
CKOUT
Vss
$\overline{CE}$
$\overline{IRQ}$
$\overline{RESET}$
PS

FIGURE 21 — MC146818 INTERFACED WITH MOTOROLA PROCESSORS

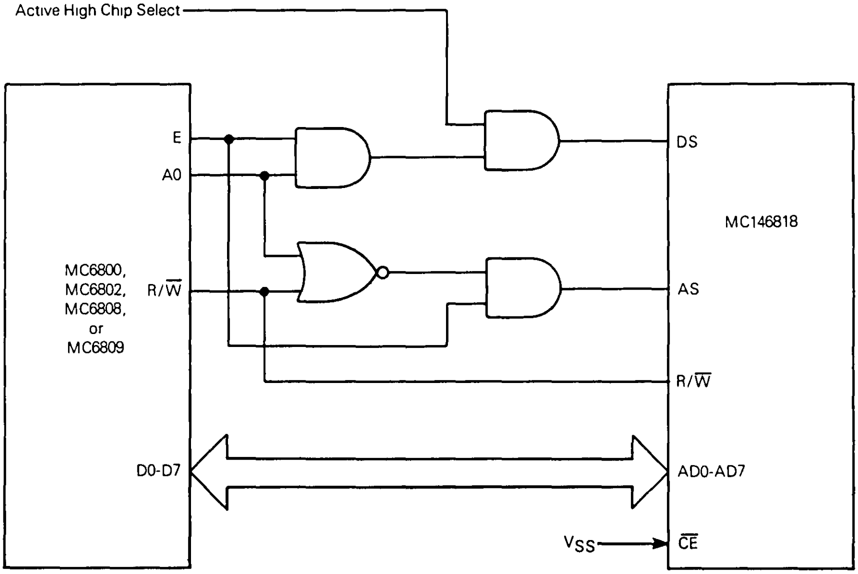

Active High Chip Select
MC6800,
MC6802,
MC6808,
or
MC6809
E
A0
R/$\overline{W}$
D0-D7
MC146818
DS
AS
R/$\overline{W}$
AD0-AD7
Vss
$\overline{CE}$

FIGURE 22 — SUBROUTINE FOR READING AND WRITING THE MC146818 WITH A NON-MULTIPLEXED BUS

```
READ     STA    RTC      Generate AS and Latch Data from ACCA
         LDAB   RTC+1    Generate DS and Get Data
         RTS

WRITE    STA    RTC      Generate AS and Latch Data from ACCA
         STAB   RTC+1    Generate DS and Store Data
         RTS
```

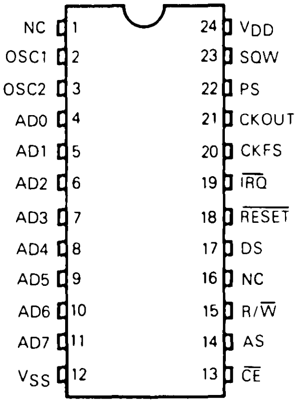
IMPORTANT NOTICES

Those devices made with date code 3N4GXXXX have the following exceptions when used in the Motorola mode of MOTEL.

1. V_{DD} = 3 to 5.25 V for operation
2. DS V_{IL} = 0.6 V Max

The falling edge of chip select should occur during the active high pulse of address strobe, only on those units with date code GC6XXXX.

PIN ASSIGNMENT


NC 1
OSC1 2
OSC2 3
AD0 4
AD1 5
AD2 6
AD3 7
AD4 8
AD5 9
AD6 10
AD7 11
VSS 12
24 VDD
23 SQW
22 PS
21 CKOUT
20 CKFS
19 IRQ
18 RESET
17 DS
16 NC
15 R/W
14 AS
13 CE

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146818A

Advance Information

Real-Time Clock Plus RAM (RTC)

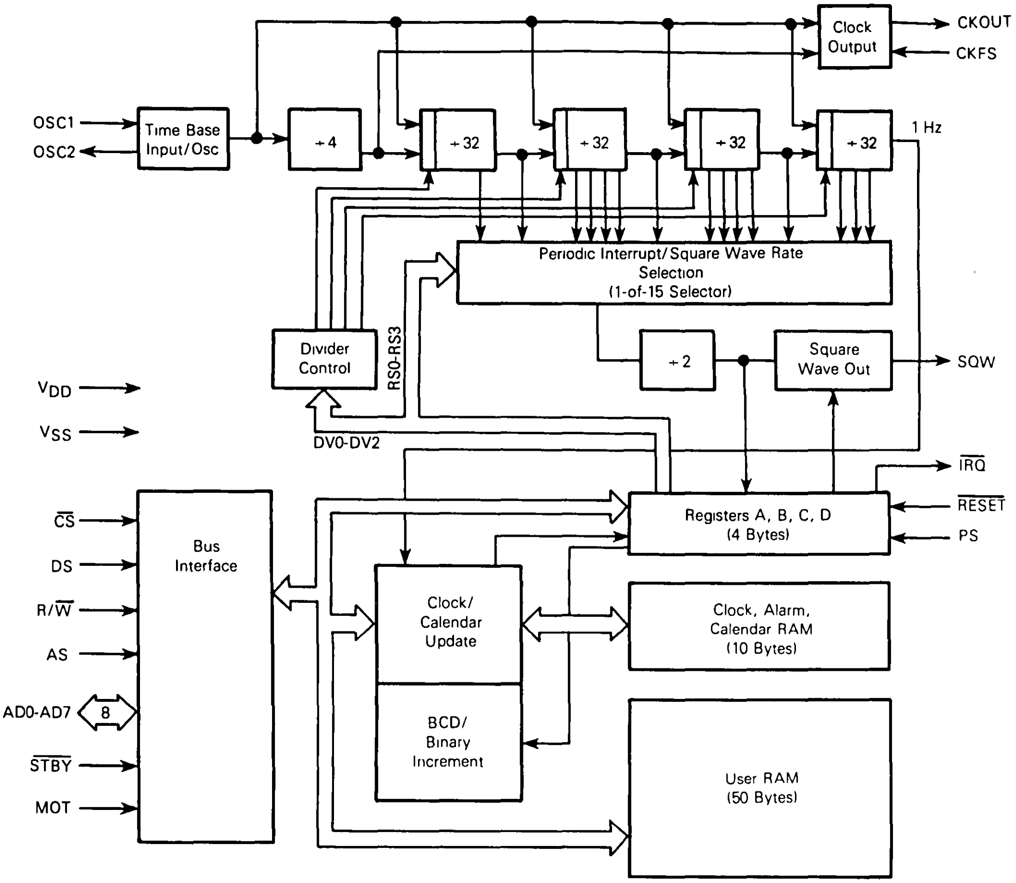
The MC146818A Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818A uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818A may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3- to 6-Volt Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 μW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- to 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- Selectable Between Motorola and Competitor Bus Timing
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{\text{IRQ}}$)
- Three Interrupts are Separately Software Maskable and Testable
  Time-of-Day Alarm, Once-per-Second to Once-per-Day
  Periodic Rates from 30.5 μs to 500 ms
  End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input At Time Base Frequency −1 or −4
- 24-Pin Dual-In-Line Package
- Quad Pack Also Available

This document contains information on a new product Specifications and information herein are subject to change without notice

FIGURE 1 — BLOCK DIAGRAM


OSC1
OSC2
Time Base Input/Osc
÷4
÷32
÷32
÷32
÷32
Clock Output
CKOUT
CKFS
1 Hz
Periodic Interrupt/Square Wave Rate Selection (1-of-15 Selector)
Divider Control
RS0-RS3
DV0-DV2
÷2
Square Wave Out
SQW
VDD
VSS
IRQ
RESET
PS
Registers A, B, C, D (4 Bytes)
CS
DS
R/W
AS
AD0-AD7
8
STBY
MOT
Bus Interface
Clock/ Calendar Update
BCD/ Binary Increment
Clock, Alarm, Calendar RAM (10 Bytes)
User RAM (50 Bytes)

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	V_{SS}−0.5 to V_{DD}+0.5	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146818A MC146818AC	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic PLCC	θ_{JA}	 120 120	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32 768	32 768	kHz
Output Voltage I_{Load} < 10 μA		V_{OL} V_{OH}	– V_{DD} – 0 1	0 1 –	V
I_{DD} – Bus Idle CKOUT = f_{osc}, C_L = 15 pF, SQW Disabled, $\overline{\text{STBY}}$ = 0 2 V, C_L (OSC2) = 10 pF f_{osc} = 32 768 kHz		I_{DD3}	–	50	μA
I_{DD} – Quiescent f_{osc} = DC, OSC1 = DC, All Other Inputs = V_{DD} – 0 2 V, No Clock		I_{DD4}	–	50	μA
Output High Voltage (I_{Load} = – 0 25 mA, All Outputs)		V_{OH}	2 7	–	V
Output Low Voltage (I_{Load} = 0 25 mA, All Outputs)		V_{OL}	–	0 3	V
Input High Voltage	$\overline{\text{STBY}}$, AD0-AD7, DS, AS, R/$\overline{\text{W}}$, $\overline{\text{CS}}$ $\overline{\text{RESET}}$, CKFS, PS, OSC1 MOT	V_{IH}	2 1 2 5 V_{DD}	V_{DD} V_{DD} V_{DD}	V
Input Low Voltage	$\overline{\text{STBY}}$, AD0-AD7, DS, AS, R/$\overline{\text{W}}$, $\overline{\text{CS}}$, CKFS, PS, $\overline{\text{RESET}}$, OSC1 MOT	V_{IL}	V_{SS} V_{SS}	0 5 V_{SS}	V
Input Current	AS, DS, R/$\overline{\text{W}}$ MOT, OSCI, $\overline{\text{CE}}$, $\overline{\text{STBY}}$, $\overline{\text{RESET}}$, CKFS, PS	I_{in}	–	± 10 ± 1	μA
Three-State Leakage	$\overline{\text{IRQ}}$, AD0-AD7	I_{TSL}	–	± 10	μA

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

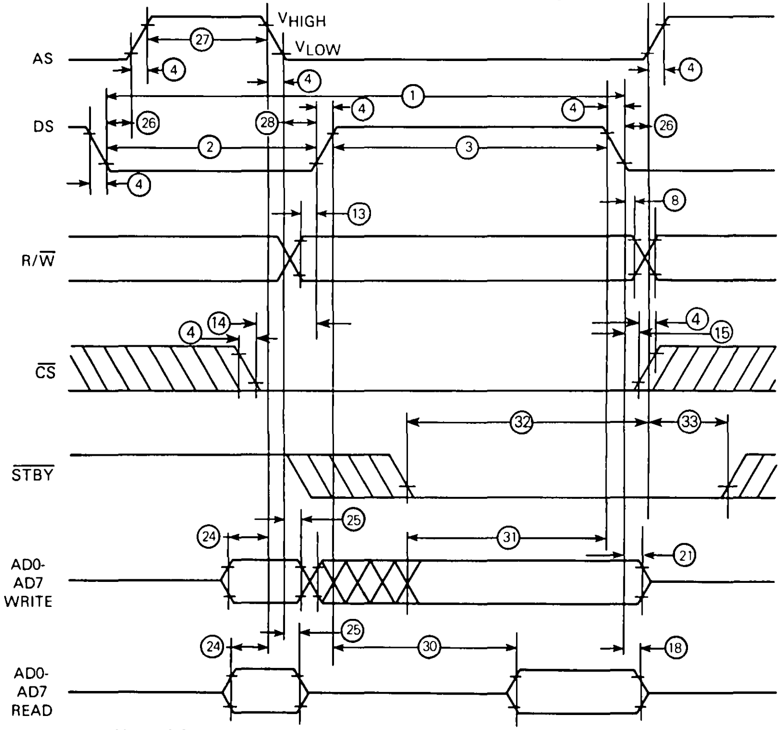
Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32 768	4194 304	kHz
Output Voltage I_{Load} < 10 μA		V_{OL} V_{OH}	– V_{DD} – 0 1	0 1 –	V
I_{DD} – Bus Idle (External Clock) CKOUT = f_{osc}, C_L = 15 pF, SQW Disabled, $\overline{\text{STBY}}$ = 0 2 V, C_L (OSC2) = 10 pF f_{osc} = 4 194304 MHz f_{osc} = 1 048516 MHz f_{osc} = 32 768 kHz		 I_{DD1} I_{DD2} I_{DD3}	 – – –	 3 800 50	 mA μA μA
I_{DD} – Quiescent f_{osc} = DC, OSC1 = DC, All Other Inputs = V_{DD} – 0 2 V, No Clock		I_{DD4}	–	50	μA
Output High Voltage (I_{Load} = – 1 6 mA, AD0-AD7, CKOUT) (I_{Load} = – 1 0 mA, SQW)		V_{OH}	4 1	–	V
Output Low Voltage (I_{Load} = 1 6 mA, AD0-AD7, CKOUT) (I_{Load} = 1 0 mA, $\overline{\text{IRQ}}$ and SQW)		V_{OL}	–	0 4	V
Input High Voltage	$\overline{\text{STBY}}$, CFKS, AD0-AD7, DS, AS, R/$\overline{\text{W}}$, $\overline{\text{CS}}$, PS $\overline{\text{RESET}}$ OSC1 MOT	V_{IH}	V_{DD} – 2 0 V_{DD} – 0 8 V_{DD} – 1 0 V_{DD}	V_{DD} V_{DD} V_{DD} V_{DD}	V
Input Low Voltage	CKFS, PS, $\overline{\text{RESET}}$, $\overline{\text{STBY}}$, AD0-AD7, DS, AS, R/$\overline{\text{W}}$, $\overline{\text{CS}}$, OSC1 MOT	V_{IL}	V_{SS} V_{SS}	0 8 V_{SS}	V
Input Current	AS, DS, R/$\overline{\text{W}}$ MOT, OSCI, $\overline{\text{CE}}$, $\overline{\text{STBY}}$, $\overline{\text{RESET}}$, CKFS, PS	I_{in}	–	± 10 ± 1	μA
Three-State Leakage	$\overline{\text{IRQ}}$, AD0-AD7	I_{TSL}	–	± 10	μA

BUS TIMING

Ident. Number	Characteristic	Symbol	$V_{DD}=3.0$ V 50 pF Load		$V_{DD}=5.0$ V ±10% 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	—	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High	PW_{EL}	1000	—	300	—	ns
3	Pulse Width, DS/E High or $\overline{RD}$/$\overline{WR}$ Low	PW_{EH}	1500	—	325	—	ns
4	Input Rise and Fall Time	t_r, t_f	—	100	—	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	—	10	—	ns
13	R/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	—	80	—	ns
14	Chip Select Setup Time Before DS, $\overline{WR}$, or $\overline{RD}$	t_{CS}	200	—	25	—	ns
15	Chip Select Hold Time	t_{CH}	10	—	0	—	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	—	0	—	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	—	50	—	ns
25	Muxed Address Hold Time	t_{AHL}	100	—	20	—	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	—	50	—	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	—	135	—	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	—	60	—	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	1300	—	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	1500	—	200	—	ns
32	$\overline{STBY}$ Setup Time before AS/ALE Rise	t_{SBS}	20	—	20	—	ns
33	$\overline{STBY}$ Hold Time after AS/ALE Fall	t_{SBH}	100	—	50	—	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.

FIGURE 2 — MC146818A BUS TIMING



Note $V_{HIGH}=V_{DD}-2.0$ V, $V_{LOW}=0.8$ V, for $V_{DD}=5.0$ V ±10% for outputs only.
$V_{HIGH}=2.0$ V, $V_{LOW}=0.5$ V, for $V_{DD}=3.0$ V for outputs only.

FIGURE 3 — BUS READ TIMING COMPETITOR MULTIPLEXED BUS

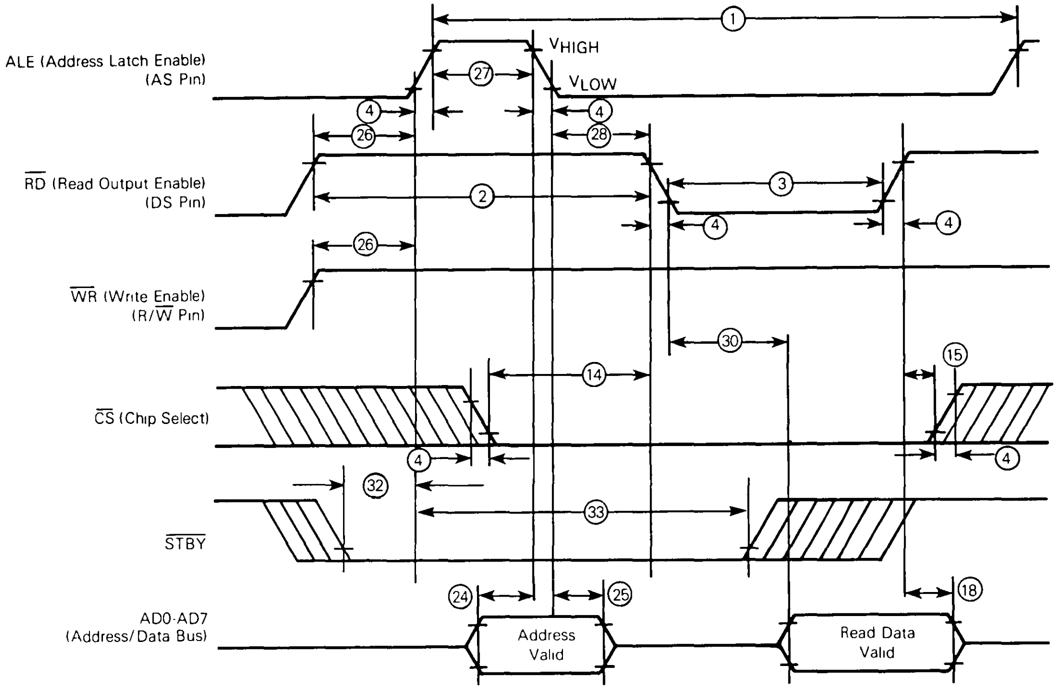
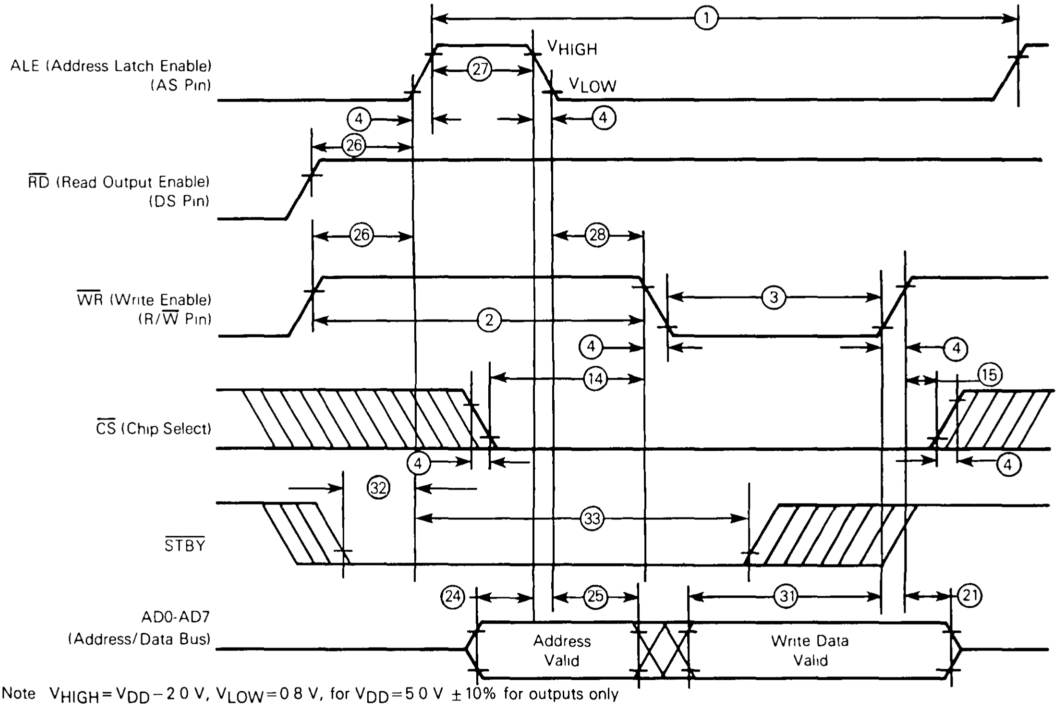

ALE (Address Latch Enable)
(AS Pin)
$\overline{RD}$ (Read Output Enable)
(DS Pin)
$\overline{WR}$ (Write Enable)
(R/$\overline{W}$ Pin)
$\overline{CS}$ (Chip Select)
$\overline{STBY}$
AD0-AD7
(Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑭
⑮
⑱
㉔
㉕
㉖
㉗
㉘
㉚
㉜
㉝
Address
Valid
Read Data
Valid

FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS

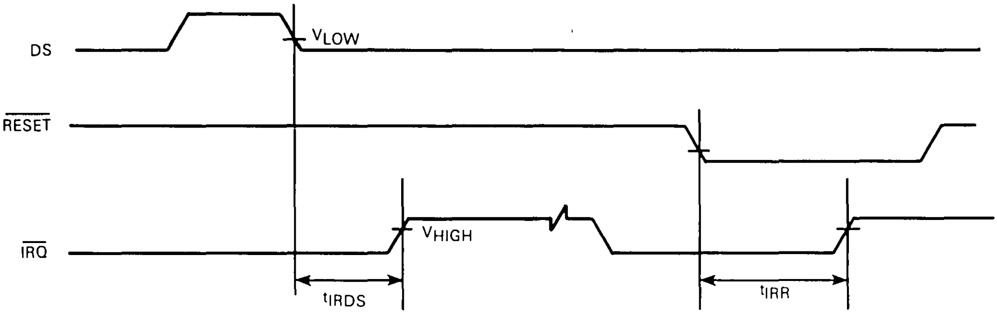

ALE (Address Latch Enable)
(AS Pin)
$\overline{RD}$ (Read Output Enable)
(DS Pin)
$\overline{WR}$ (Write Enable)
(R/$\overline{W}$ Pin)
$\overline{CS}$ (Chip Select)
$\overline{STBY}$
AD0-AD7
(Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑭
⑮
㉑
㉔
㉕
㉖
㉗
㉘
㉛
㉜
㉝
Address
Valid
Write Data
Valid

Note $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10% for outputs only
$V_{HIGH} = 2.0$ V, $V_{LOW} = 0.5$ V, for $V_{DD} = 3.0$ V for outputs only

TABLE 1 – SWITCHING CHARACTERISTICS ($V_{SS}=0$ Vdc, $T_A=T_L$ to T_H)

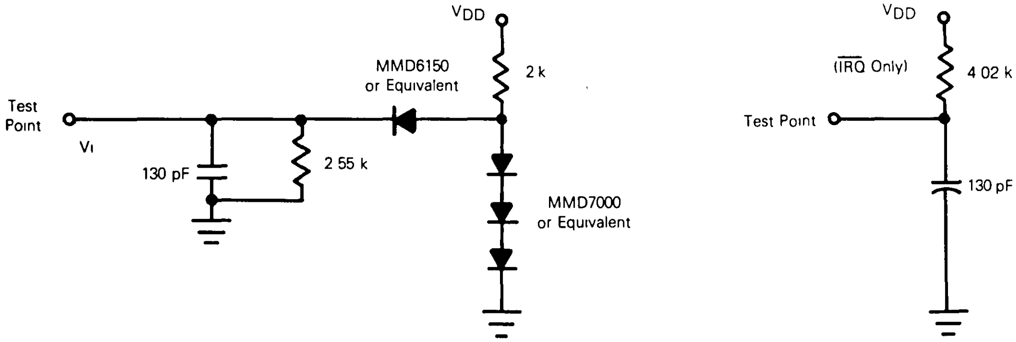
Description	Symbol	$V_{DD}=3.0$ Vdc		$V_{DD}=5.0$ Vdc ±10%		Unit
		Min	Max	Min	Max	
Oscillator Startup	t_{RC}	—	300	—	100	ms
Reset Pulse Width	t_{RWL}	25	—	5	—	µs
Reset Delay Time	t_{RLH}	25	—	5	—	µs
Power Sense Pulse Width	t_{PWL}	25	—	5	—	µs
Power Sense Delay Time	t_{PLH}	25	—	5	—	µs
$\overline{\text{IRQ}}$ Release from DS	t_{IRDS}	—	10	—	2	µs
$\overline{\text{IRQ}}$ Release from $\overline{\text{RESET}}$	t_{IRR}	—	10	—	2	µs
VRT Bit Delay	t_{VRTD}	—	10	—	2	µs

FIGURE 5 — $\overline{\text{IRQ}}$ RELEASE DELAY


DS
V_{LOW}
$\overline{\text{RESET}}$
$\overline{\text{IRQ}}$
V_{HIGH}
t_{IRDS}
t_{IRR}

NOTE. $V_{HIGH}=V_{DD}-2.0$ V, $V_{LOW}=0.8$ V, for $V_{DD}=5.0$ V ± 10%

FIGURE 6 — TTL EQUIVALENT TEST LOAD


V_{DD}
MMD6150
or Equivalent
2 k
Test Point
V_i
130 pF
2.55 k
MMD7000
or Equivalent
V_{DD}
($\overline{\text{IRQ}}$ Only)
4.02 k
Test Point
130 pF

All Outputs Except OSC2 (See Figure 10)

FIGURE 7 — POWER-UP

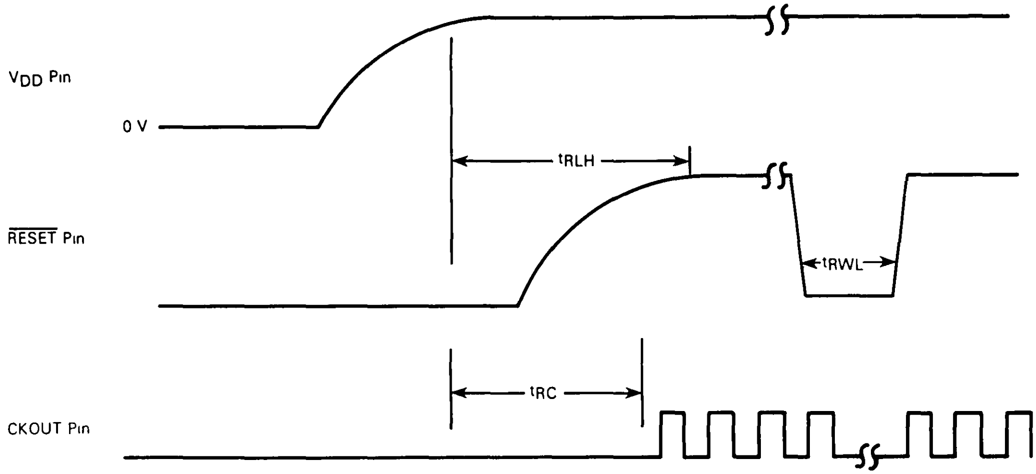
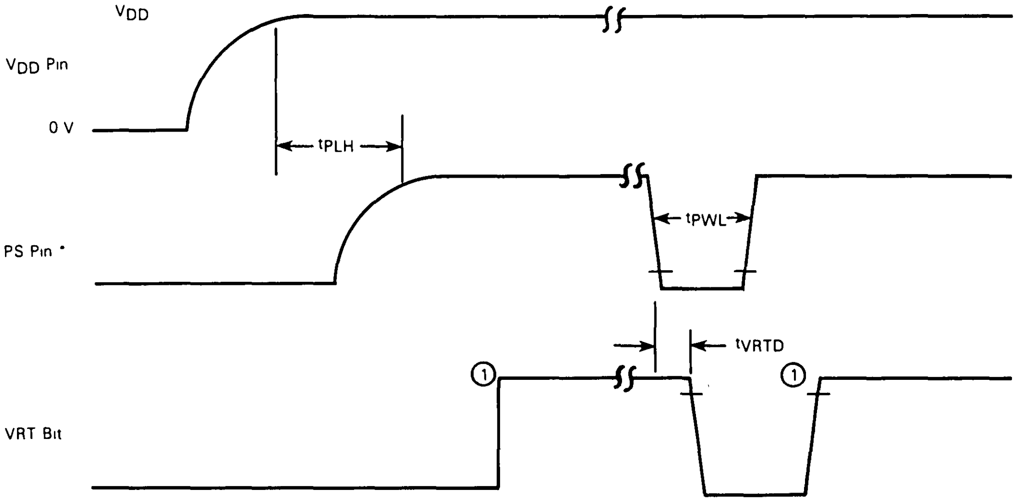

VDD Pin
0 V
tRLH
RESET Pin
tRWL
tRC
CKOUT Pin

FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT


VDD
VDD Pin
0 V
tPLH
PS Pin *
tPWL
tVRTD
①
①
VRT Bit

① The VRT bit is set to a "1" by reading Register d The VRT bit can only be cleared by pulling the PS pin low (see **REGISTER D ($0D)**).

SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the MC146818A Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins, V_{DD} being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

MOT—MOTEL

The MOT pin offers flexibility when choosing bus type. When tied to V_{DD}, Motorola timing is used. When tied to V_{SS}, competitor timing is used. The MOT pin must be hardwired to the V_{DD} or V_{SS} supply and cannot be switched during operation of the MC146818A.

OSC1, OSC2 — TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 9. The internal time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz, 1.048576 MHz or 32.768 kHz frequencies. The crystal connections are shown in Figure 10 and the crystal characteristics in Figure 11.

CKOUT — CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS — CLOCK OUT FREQUENCY SELECT, INPUT

When the CKFS pin is tied to V_{DD}, it causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is tied to V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

TABLE 2 — CLOCK OUTPUT FREQUENCIES

Time Base (OSC1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4 194304 MHz	High	4 194304 MHz
4 194304 MHz	Low	1 048576 MHz
1 048576 MHz	High	1 048576 MHz
1 048576 MHz	Low	262 144 kHz
32 768 kHz	High	32 768 kHz
32 768 kHz	Low	8.192 kHz

SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages. The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B.

AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the MC146818A since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the MC146818A latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146818A outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to the high-impedance state) when DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case.

AS — MULTIPLEXED ADDRESS STROBE, INPUT

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the MC146818A. The address latch within the MC146818A operates independent of $\overline{CS}$.

DS — DATA STROBE OR READ, INPUT

The DS pin has two interpretations via the MOTEL circuit. When emanating from a Motorola type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from the competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

R/$\overline{W}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{W}$ pin in one of two ways. When a Motorola type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ during DS.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor type processors. The MOTEL circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

$\overline{CS}$ — CHIP SELECT, INPUT

The chip-select ($\overline{CS}$) signal must be asserted (low) for a bus cycle in which the MC146818A is to be accessed. $\overline{CS}$ is not latched and must be stable during DS and AS (Motorola case of MOTEL) and during $\overline{RD}$ and $\overline{WR}$. When $\overline{CS}$ is not used, it should be grounded. (See Figure 20).

FIGURE 9 — EXTERNAL TIME-BASE CONNECTION

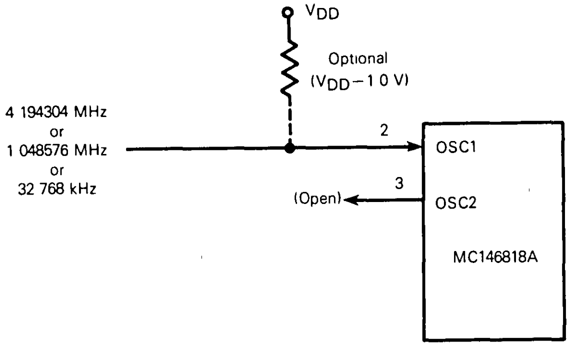
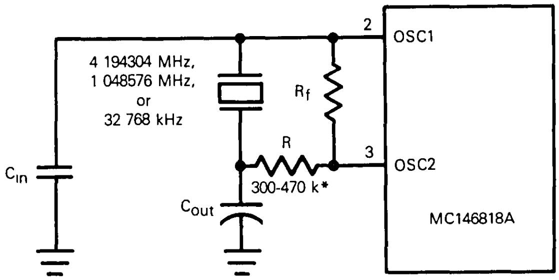

VDD
Optional
(VDD − 1 0 V)
4 194304 MHz
or
1 048576 MHz
or
32 768 kHz
2
3
(Open)
OSC1
OSC2
MC146818A

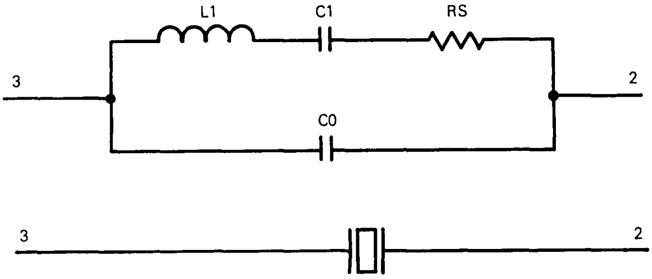
FIGURE 10 — CRYSTAL OSCILLATOR CONNECTION


2
OSC1
4 194304 MHz,
1 048576 MHz,
or
32 768 kHz
Rf
R
3
OSC2
Cin
300-470 k*
Cout
MC146818A

*32 768 kHz Only — Consult Crystal Manufacturer's Specification

FIGURE 11 — CRYSTAL PARAMETERS

Crystal Equivalent Circuit


L1
C1
RS
3
2
C0
3
2

f_{osc}	4.194304 MHz	1.048576 MHz	32.768 kHz
RS (Maximum)	75 Ω	700 Ω	50 k
C0 (Maximum)	7 pF	5 pF	1 7 pF
C1	0 012 pF	0 008 pF	0 003 pF
Q	50 k	35 k	30 k
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
R	—	—	300-470 k
R_f	10 M	10 M	22 M

$\overline{\text{IRQ}}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{\text{IRQ}}$ pin is an active low output of the MC146818A that may be used as an interrupt input to a processor. The $\overline{\text{IRQ}}$ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the $\overline{\text{IRQ}}$ pin, the processor program normally reads Register C. The $\overline{\text{RESET}}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{\text{IRQ}}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{\text{IRQ}}$ bus with one pullup at the processor.

$\overline{\text{RESET}}$ — RESET, INPUT

The $\overline{\text{RESET}}$ pin does not affect the clock, calendar, or RAM functions. On powerup, the $\overline{\text{RESET}}$ pin must be held low for the specified time, t_{RLH}, in order to allow the power supply to stabilize. Figure 12 shows a typical representation of the $\overline{\text{RESET}}$ pin circuit.

When $\overline{\text{RESET}}$ is low the following occurs.

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Alarm Interrupt Enable (AIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible.
h) Alarm Interrupt Flag (AF) bit is cleared to zero,
i) $\overline{\text{IRQ}}$ pin is in high-impedance state,
j) Square Wave output Enable (SQWE) bit is cleared to zero, and
k) Standby Input Enabled if AS is low.

$\overline{\text{STBY}}$ — STANDBY

The $\overline{\text{STBY}}$ pin, when active, prevents access to the MC146818A making it ideal for battery back-up applications. Standby operation incorporates a transparent latch. After data strobe (DS) goes low ($\overline{\text{RD}}$ or $\overline{\text{WR}}$ rises), $\overline{\text{STBY}}$ is recognized as a valid signal.

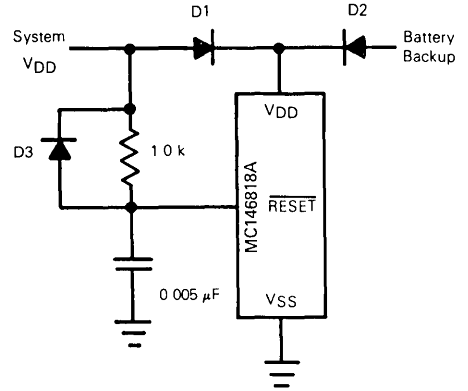
The $\overline{\text{STBY}}$ signal is totally asynchronous. Its transparent latch is opened by the falling edge of DS (rising edge of $\overline{\text{RD}}$ or $\overline{\text{WR}}$) and clocked by the rising edge of AS (ALE). Therefore, for $\overline{\text{STBY}}$ to be recognized , DS and AS should occur in pairs. When $\overline{\text{STBY}}$ goes low before the falling edge of DS (rising edge of $\overline{\text{WR}}$ or $\overline{\text{RD}}$), the current cycle is completed at that edge, and the next cycle will not be executed. A low on the $\overline{\text{RESET}}$ pin enables the standby input.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low, the VRT bit is cleared to zero.

When using the VRT feature during powerup, the PS pin must be externally held low for the specified t_{PLH} time. As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. PS must go high after powerup to allow the VRT bit to be set by a read of Register D.

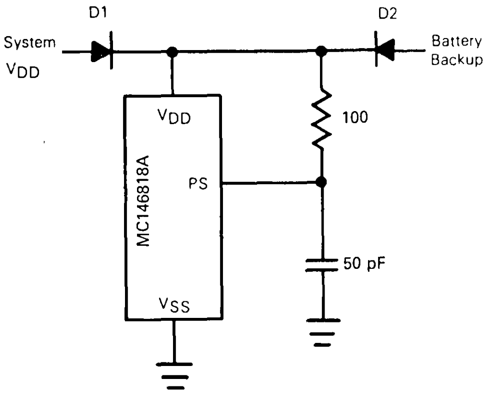
FIGURE 12—TYPICAL POWERUP DELAY CIRCUIT FOR RESET



D1 = MBD701 (Schottky) or Equivalent
D2 = D3 = 1N4148 or Equivalent

Note If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements

FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR POWER SENSE



D1 = MBD701 (Schottky) or Equivalent
D2 = 1N4148 or Equivalent

POWER-DOWN CONSIDERATIONS

In most systems, the MC146818A must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The $\overline{\text{STBY}}$ pin controls all bus inputs (R/$\overline{\text{W}}$, DS, AS, AD0-AD7). $\overline{\text{STBY}}$, when asserted, disallows any unintended modification of the RTC data by the bus. $\overline{\text{STBY}}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part

ADDRESS MAP

Figure 14 shows the address map of the MC146818A The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes All 64 bytes are directly readable and writable by the processor program except for the following 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD)

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD The SET bit may now be cleared to allow updates Once initialized the real-time clock makes all updates in the selected data mode The data mode cannot be changed without reinitializing the 10 data bytes

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1"

The time, calendar, and alarm bytes are not always accessible by the processor program. Once per second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition If any of the 10 bytes are read at this time, the data outputs are undefined The update lockout time is 248 µs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 µs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes The "don't care" codes in all three alarm bytes create an interrupt every second.

FIGURE 14 — ADDRESS MAP

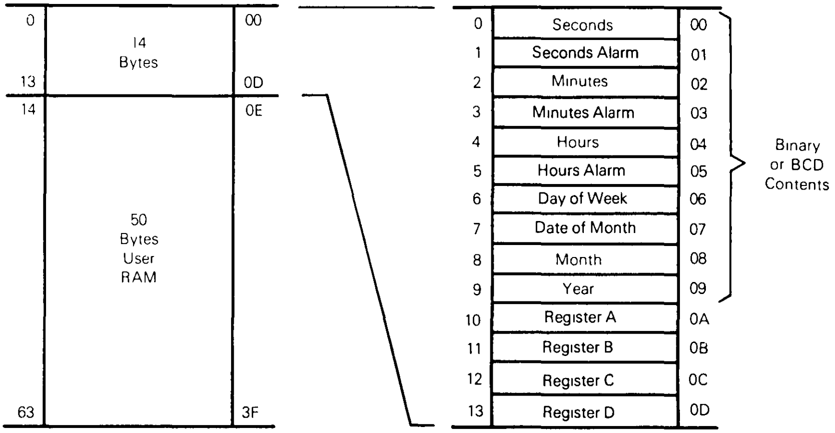


TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

Address Location	Function	Decimal Range	Range		Example*	
			Binary Data Mode	BCD Data Mode	Binary Data Mode	BCD Data Mode
0	Seconds	0-59	$00-$3B	$00-$59	15	21
1	Seconds Alarm	0-59	$00-$3B	$00-$59	15	21
2	Minutes	0-59	$00-$3B	$00-$59	3A	58
3	Minutes Alarm	0-59	$00-$3B	$00-$59	3A	58
4	Hours (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	$00-$17	$00-$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	$00-$17	$00-23	05	05
6	Day of the Week Sunday = 1	1-7	$01-$07	$01-$07	05	05
7	Date of the Month	1-31	$01-$1F	$01-$31	0F	15
8	Month	1-12	$01-$0C	$01-$12	02	02
9	Year	0-99	$00-$63	$00-$99	4F	79

*Example: 5 58 21 Thursday 15 February 1979 (time is AM)

STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the MC146818A. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional MC146818As may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. The high-order bit of the seconds byte, bit 7 of Register A, and all bits of Registers C and D cannot effectively be used as general purpose RAM.

INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 µs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{\text{IRQ}}$ pin is immediately activiated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts

When an interrupt event occurs, a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes When the software detects that the flag is set, it is an indication to software that the "interrupt"event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held after the read cycle. One, two or three flag bits may be found to be set when Register C is used The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{\text{IRQ}}$ pin is asserted low. $\overline{\text{IRQ}}$ is asserted as long as at least one of the three interrupt sources has its flag and enables bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{\text{IRQ}}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7

(IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

DIVIDER STAGES

The MC146818A has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The divers are controlled by three divider bus (DV2, DV1, and DV0) in Register A.

DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32 768 kHz). The divider chain may be held at reset, which allows precision setting of the time, When the divider is changed from reset to an operating time base, the first update cycle is one-half second later. The divider-control bits are also used to facilitate testing the MC146818A.

SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes	–	N=0
1.048576 MHz	0	0	1	Yes	–	N=2
32.768 kHz	0	1	0	Yes	–	N=7
Any	1	1	0	No	Yes	–
Any	1	1	1	No	Yes	–

Note Other combinations of divider bits are used for test purposes only

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Select Bits Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate t_{PI}	SQW Output Frequency	Periodic Interrupt Rate t_{PI}	SQW Output Frequency
0	0	0	0	None	None	None	None
0	0	0	1	30.517 µs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 µs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 µs	8.192 kHz	122.070 µs	8.192 kHz
0	1	0	0	244.141 µs	4.096 kHz	244.141 µs	4.096 kHz
0	1	0	1	488.281 µs	2.048 kHz	488.281 µs	2.048 kHz
0	1	1	0	976.562 µs	1.024 kHz	976.562 µs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once per second to once per day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit in Register B. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial recieve bits or bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

UPDATE CYCLE

The MC146818A executes an update cycle once per second, assuming one of the proper time bases is in place, the DV0-DV2 divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 μs while a 32.768 kHz time base update cycle takes 1984 μs. During the update cycle, the time, calendar, and alarm bytes are not accessible by the processor program. The MC146818A protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete, the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods, it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

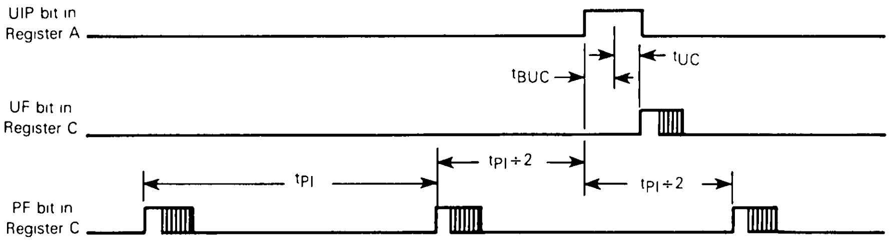
The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transferred to continuously available RAM. Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once per second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 μs later. Therefore, if a low is read on the UIP bit, the user has at least 244 μs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 μs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 15). Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to ensure that data is not read during the update cycle.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

FIGURE 15 — UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIP



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62 5 ms, etc per Table 5)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

REGISTERS

The MC146818A has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

REGISTER A ($0A)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

Read/Write Register except UIP

UIP — The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1", the update cycle is in progress or will soon begin. When UIP is a "0", the update cycle is not in progress and will not be for at least 244 μs (for all time bases) This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibits any update cycle and then clears the UIP status bit.

TABLE 6 — UPDATE CYCLE TIMES

UIP Bit	Time Base (OSC1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4.194304 MHz	248 μs	–
1	1.048576 MHz	248 μs	–
1	32.768 kHz	1984 μs	–
0	4.194304 MHz	–	244 μs
0	1.048576 MHz	–	244 μs
0	32.768 kHz	–	244 μs

DV2, DV1, DV0 — Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed, the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

RS3, RS2, RS1, RS0 — The four rate selection bits select one of 15 tapes on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{\text{RESET}}$.

REGISTER B ($0B)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

Read/Write Register

SET — When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is a read/write bit which is not modified by $\overline{\text{RESET}}$ or internal functions of the MC146818A.

PIE — The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{\text{IRQ}}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{\text{IRQ}}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal MC146818A functions, but is cleared to "0" by a $\overline{\text{RESET}}$.

AIE — The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) bit in Register C to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code by binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{\text{IRQ}}$ signal. The $\overline{\text{RESET}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE — The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flag (UF) bit in Register C to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

SQWE — When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

DM — The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12 — The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

DSE — The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

REGISTER C ($0C)

MSB LSB

b7	b6	b5	b4	b3	b	b1	b0
IRQF	PF	AF	UF	0	0	0	0

Read-Only Register

IRQF — The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"
AF = AIE = "1"
UF = UIE = "1"

i.e., IRQF = PF•PIE + AF•AIE + UF•UIE

Any time the IRQF bit is a "1", the IRQ pin is driven low. All flag bits are cleared after Register C is read by the program or when the RESET pin is low.

PF — The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an IRQ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a RESET or a software read of Register C.

AF — A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the IRQ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A RESET or a read of Register C clears AF.

UF — The update-ended interrupt flag (UF) bit is set after each update cycle. when the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting IRQ. UF is cleared by a Register C read or a RESET.

b3 TO b0 — The unused bits of Status Register 1 are read as "0's". They can not be written.

REGISTER D ($0D)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
VRT	0	0	0	0	0	0	0

Read Only Register

VRT — The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read only bit which is not modified by the RESET pin. The VRT bit can only be set by reading Register D.

b6 TO b0 — The remaining bits of Register D are unused. They cannot be written, but are always read as "0's."

TYPICAL INTERFACING

The MC146818A is best suited for use with microprocessors which generate an address-then-data multiplexed bus Figures 16 and 17 show typical interfaces to bus-compatible processors. These interfaces assume that the address decoding can be done quickly. However, if standard metalgate CMOS gates are used, the CS setup time may be violated Figure 18 illustrates an alternative method of chip selection which will accommodate such slower decoding

The MC146818A can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 19. Non-multiplexed bus microprocessors can be interfaced with additional support

There is one method of using the multiplexed bus MC146818A with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together.

An example using either the Motorola MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 20. When the MC146818A is I/O mapped as shown in Figures 19 and 20, the AS and DS inputs should be left in a low state when the part is not being accessed Refer to the STBY pin description for the conditions which must be met before STBY can be recognized.

Figure 21 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data.

Accumulator A: The address of the RTC to be accessed
Accumulator B: Write: The data to be written
Read: The data read from the RTC

The RTC is mapped to two consecutive memory locations — RTC and RTC+1 as shown in Figure 20

FIGURE 16 — MC146818A INTERFACED WITH MOTOROLA COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

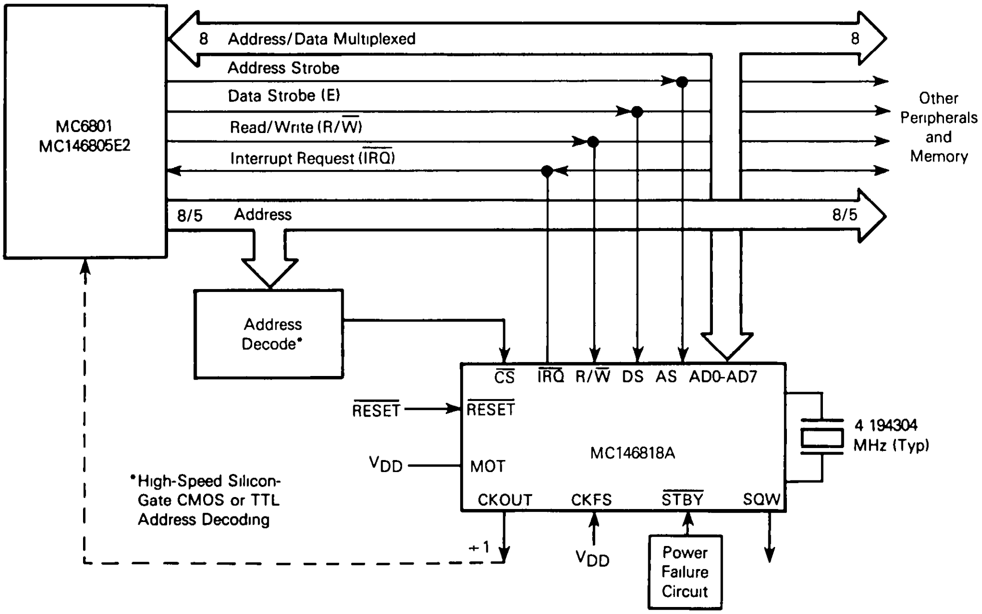


FIGURE 17 — MC146818A INTERFACED WITH COMPETITOR COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

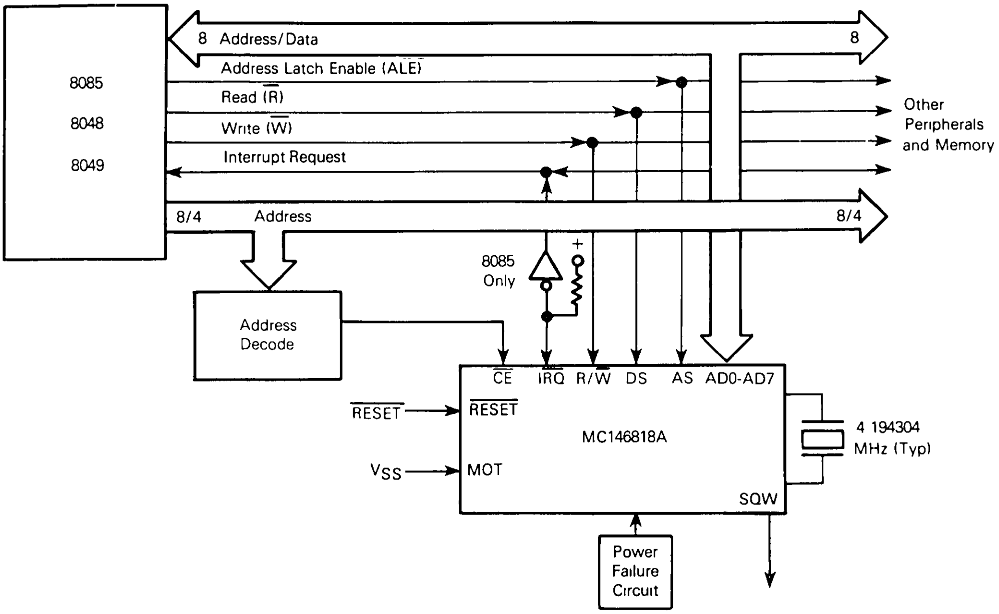
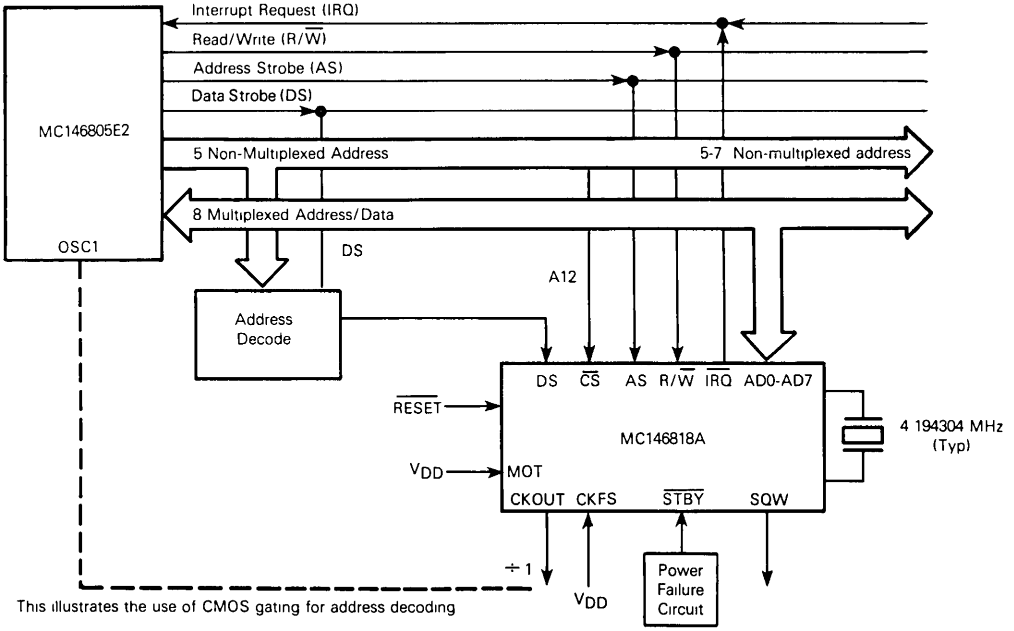
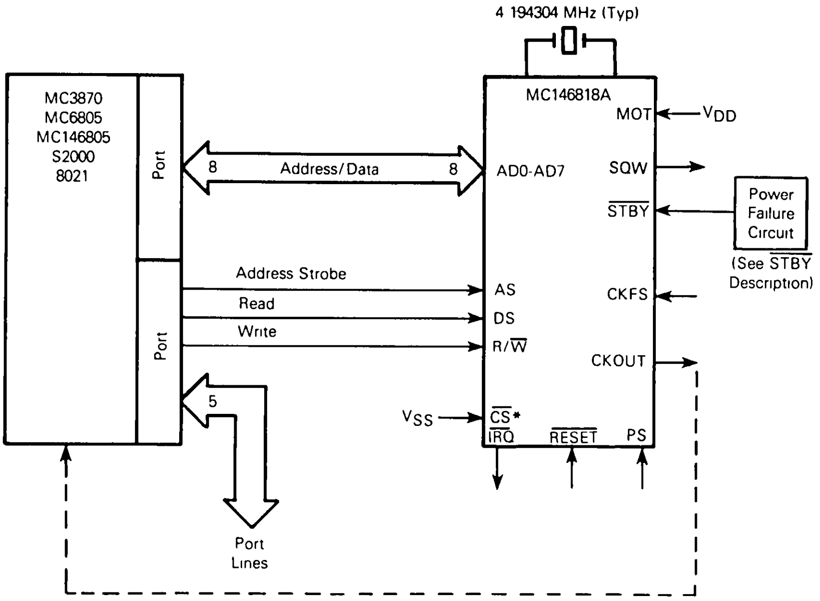

8085
8048
8049
8 Address/Data
8
Address Latch Enable (ALE)
Read ($\overline{R}$)
Write ($\overline{W}$)
Interrupt Request
Other Peripherals and Memory
8/4
Address
8/4
8085 Only
+
Address Decode
$\overline{CE}$
$\overline{IRQ}$
R/$\overline{W}$
DS
AS
AD0-AD7
$\overline{RESET}$
$\overline{RESET}$
MC146818A
4 194304 MHz (Typ)
V_{SS}
MOT
SQW
Power Failure Circuit

FIGURE 18 — MC146818A INTERFACE WITH MC146805E2 CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING


MC146805E2
OSC1
Interrupt Request ($\overline{IRQ}$)
Read/Write (R/$\overline{W}$)
Address Strobe (AS)
Data Strobe (DS)
5 Non-Multiplexed Address
5-7 Non-multiplexed address
8 Multiplexed Address/Data
DS
A12
Address Decode
DS
$\overline{CS}$
AS
R/$\overline{W}$
$\overline{IRQ}$
AD0-AD7
$\overline{RESET}$
MC146818A
4 194304 MHz (Typ)
V_{DD}
MOT
CKOUT
CKFS
$\overline{STBY}$
SQW
÷1
V_{DD}
Power Failure Circuit

This illustrates the use of CMOS gating for address decoding

FIGURE 19 — MC146818A INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER


4 194304 MHz (Typ)
MC3870
MC6805
MC146805
S2000
8021
Port
Port
8
Address/Data
8
MC146818A
MOT
VDD
AD0-AD7
SQW
STBY
Power Failure Circuit
(See STBY Description)
Address Strobe
Read
Write
AS
DS
R/W
CKFS
CKOUT
5
VSS
CS*
IRQ
RESET
PS
Port Lines

*NOTE. $\overline{CS}$ can be controlled by a port pin (if available)

FIGURE 20 — MC146818A INTERFACED WITH MOTOROLA PROCESSORS

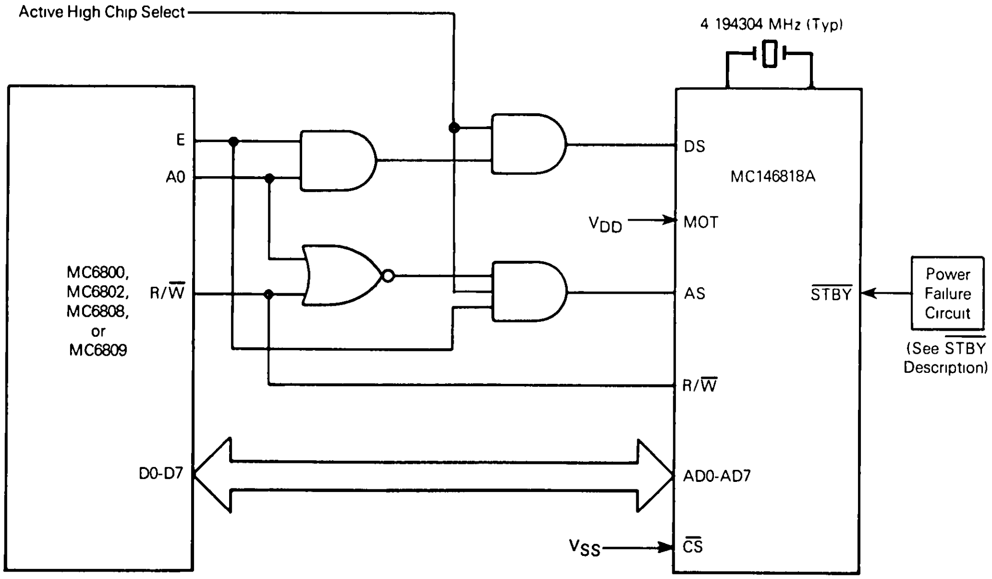
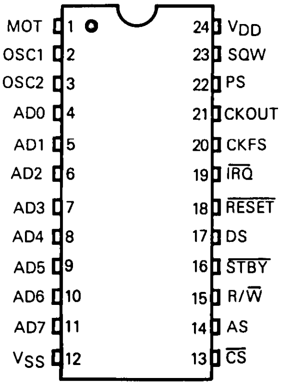

Active High Chip Select
4 194304 MHz (Typ)
E
A0
MC6800,
MC6802,
MC6808,
or
MC6809
R/W
D0-D7
DS
MC146818A
VDD
MOT
AS
STBY
Power Failure Circuit
(See STBY Description)
R/W
AD0-AD7
VSS
CS

FIGURE 21 — SUBROUTINE FOR READING AND WRITING THE MC146818A WITH A NON-MULTIPLEXED BUS

```
READ     STA     RTC        Generate AS and Latch Data from ACCA
         LDAB    RTC+1      Generate DS and Get Data
         RTS

WRITE    STA     RTC        Generate AS and Latch Data from ACCA
         STAB    RTC+1      Generate DS and Store Data
         RTS
```

MECHANICAL DATA

PIN ASSIGNMENTS


MOT 1
OSC1 2
OSC2 3
AD0 4
AD1 5
AD2 6
AD3 7
AD4 8
AD5 9
AD6 10
AD7 11
V_{SS} 12
24 V_{DD}
23 SQW
22 PS
21 CKOUT
20 CKFS
19 $\overline{IRQ}$
18 $\overline{RESET}$
17 DS
16 $\overline{STBY}$
15 R/$\overline{W}$
14 AS
13 $\overline{CS}$

NOTE: Pin assignments are the same for both the dual-in-line and quad (PLCC) packages.

MC6821

Peripheral Interface Adapter (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 Family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation

3

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0 3 to +7 0	V
Input Voltage	V_{in}	−0 3 to +7 0	V
Operating Temperature Range MC6821, MC68A21, MC68B21 MC6821C, MC68A21C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		100	
Cerdip		60	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$

Unused inputs must always be tied to an appropriate logic voltage level (e g , either V_{SS} or V_{CC})

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K - (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5 0 Vdc ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R/$\overline{W}$, Enable, $\overline{RESET}$, RS0, RS1, CS0, CS1, CS2)					
Input High Voltage	V_{IH}	V_{SS}+2 0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}−0 3	–	V_{SS}+0 8	V
Input Leakage Current (V_{in} = 0 to 5 25 V)	I_{in}	–	1 0	2 5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)	C_{in}	–	–	7 5	pF
INTERRUPT OUTPUTS ($\overline{IRQA}$, $\overline{IRQB}$)					
Output Low Voltage (I_{Load} = 3 2 mA)	V_{OL}	–	–	V_{SS}+0 4	V
Hi-Z Output Leakage Current	I_{OZ}	–	1 0	10	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)	C_{out}	–	–	5 0	pF
DATA BUS (D0-D7)					
Input High Voltage	V_{IH}	V_{SS}+2 0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}−0 3	–	V_{SS}+0 8	V
Hi-Z Input Leakage Current (V_{in} = 0 4 to 2 4 V)	I_{IZ}	–	2 0	10	μA
Output High Voltage (I_{Load} = −205 μA)	V_{OH}	V_{SS}+2 4	–	–	V
Output Low Voltage (I_{Load} = 1 6 mA)	V_{OL}	–	–	V_{SS}+0 4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)	C_{in}	–	–	12 5	pF

DC ELECTRICAL CHARACTERISTICS (Continued)

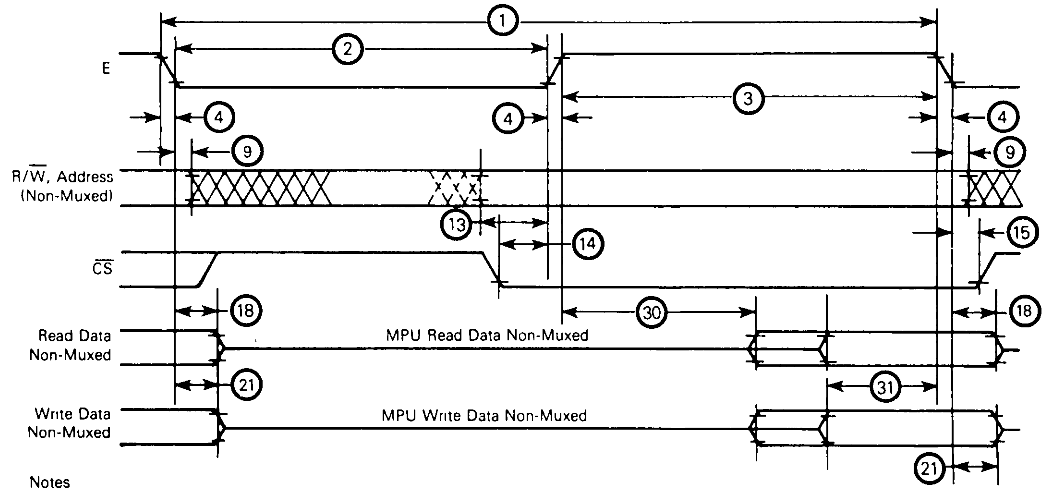
Characteristic		Symbol	Min	Typ	Max	Unit
PERIPHERAL BUS (PA0-PA7, PB0-PB7, CA1, CA2, CB1, CB2)						
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$, CA1, CB1, Enable	I_{in}	–	1 0	2 5	μA
Hi-Z Input Leakage Current (V_{in} = 0 4 to 2 4 V)	PB0-PB7, CB2	I_{IZ}	–	2 0	10	μA
Input High Current (V_{IH} = 2.4 V)	PA0-PA7, CA2	I_{IH}	– 200	– 400	–	μA
Darlington Drive Current (V_O = 1 5 V)	PB0-PB7, CB2	I_{OH}	– 1 0	–	– 10	mA
Input Low Current (V_{IL} = 0.4 V)	PA0-PA7, CA2	I_{IL}	–	– 1.3	– 2 4	mA
Output High Voltage (I_{Load} = – 200 μA) (I_{Load} = – 10 μA)	 PA0-PA7, PB0-PB7, CA2, CB2 PA0-PA7, CA2	V_{OH}	 V_{SS} + 2 4 V_{CC} – 1 0	 – –	 – –	V
Output Low Voltage (I_{Load} = 3 2 mA)		V_{OL}	–	–	V_{SS} + 0 4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)		C_{in}	–	–	10	pF
POWER REQUIREMENTS						
Internal Power Dissipation (Measured at T_L = 0°C)		P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1 0	10	0 67	10	0 5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance)

FIGURE 1 — BUS TIMING



Notes

1 Voltage levels shown are $V_L \leq 0\ 4$ V, $V_H \geq 2\ 4$ V, unless otherwise specified
2 Measurement points shown are 0 8 V and 2 0 V, unless otherwise specified

PERIPHERAL TIMING CHARACTERISTICS (V_{CC} = 5.0 V +5%, V_{SS} = 0 V, $T_A = T_L$ to T_H, unless otherwise specified)

Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit	Reference Fig No
		Min	Max	Min	Max	Min	Max		
Data Setup Time	t_{PDS}	200	—	135	—	100	—	ns	6
Data Hold Time	t_{PDH}	0	—	0	—	0	—	ns	6
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	—	1.0	—	0.670	—	0.500	µs	3, 7, 8
Delay Time, Enable Negative Transition t CA2 Positive Transition	t_{RS1}	—	1.0	—	0.670	—	0.500	µs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	—	1.0	—	1.0	—	1.0	µs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	t_{RS2}	—	2.0	—	1.35	—	1.0	µs	3, 8
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	—	1.0	—	0.670	—	0.5	µs	3, 9, 10
Delay Time, Enable Negative Transition to CMOS Data Valid PA0-PA7, CA2	t_{CMOS}	—	2.0	—	1.35	—	1.0	µs	4, 9
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	—	1.0	—	0.670	—	0.5	µs	3, 11, 12
Delay Time, Data Valid to CB2 Negative Transition	t_{DD}	20	—	20	—	20	—	ns	3, 10
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	—	1.0	—	0.670	—	0.5	µs	3, 11
Control Output Pulse Width, CA2/CB2	PW_{CT}	500	—	375	—	250	—	ns	3, 7, 11
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	—	1.0	—	1.0	—	1.0	µs	12
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	—	2.0	—	1.35	—	1.0	µs	3, 12
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	t_{IR}	—	1.60	—	1.10	—	0.85	µs	5, 14
Interrupt Response Time	t_{RS3}	—	1.0	—	1.0	—	1.0	µs	5, 13
Interrupt Input Pulse Time	PW_I	500	—	500	—	500	—	ns	13
$\overline{RESET}$ Low Time*	t_{RL}	1.0	—	0.66	—	0.5	—	µs	15

FIGURE 2 — BUS TIMING TEST LOADS

(D0-D7)

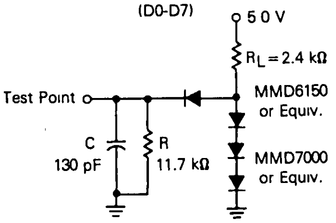
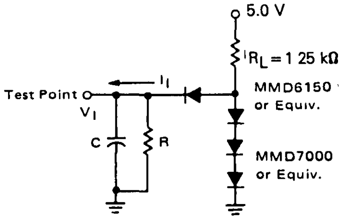


FIGURE 3 — TTL EQUIVALENT TEST LOAD

(PA0–PA7, PB0–PB7, CA2, CB2)



C = 30 pF, R = 12 k

FIGURE 4 — CMOS EQUIVALENT TEST LOAD

(PA0-PA7, CA2)

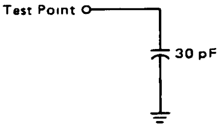


FIGURE 5 — NMOS EQUIVALENT TEST LOAD

($\overline{IRQ}$ Only)

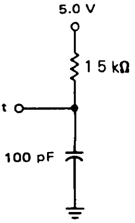


FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES (Read Mode)

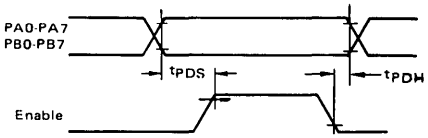
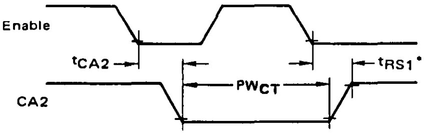

PA0-PA7
PB0-PB7
tPDS
tPDH
Enable

FIGURE 7 — CA2 DELAY TIME (Read Mode; CRA-5 = CRA3 = 1, CRA-4 = 0)


Enable
tCA2
PWCT
tRS1*
CA2

*Assumes part was deselected during the previous E pulse.

FIGURE 8 — CA2 DELAY TIME (Read Mode; CRA-5 = 1, CRA-3 = CRA-4 = 0)

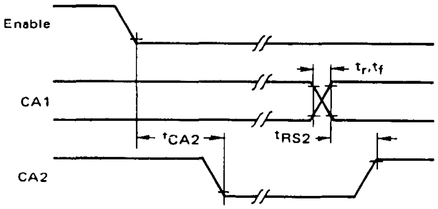

Enable
tr, tf
CA1
tCA2
tRS2
CA2

FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES (Write Mode; CRA-5 = CRA-3 = 1, CRA-4 = 0)

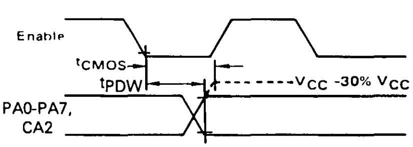
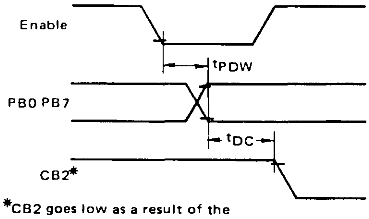
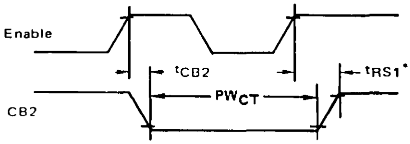

Enable
tCMOS
tPDW
VCC -30% VCC
PA0-PA7,
CA2

FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES (Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)


Enable
tPDW
PB0 PB7
tDC
CB2*

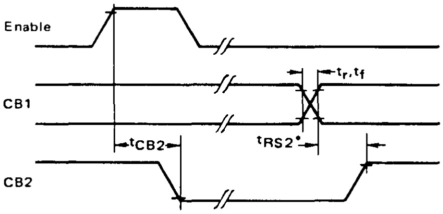
*CB2 goes low as a result of the positive transition of Enable

FIGURE 11 — CB2 DELAY TIME (Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)


Enable
tCB2
tRS1*
PWCT
CB2

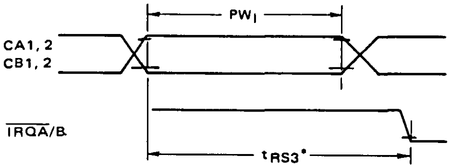
*Assumes part was deselected during the previous E pulse.

FIGURE 12 — CB2 DELAY TIME (Write Mode; CRB-5 = 1, CRB-3 = CRB-4 = 0)


Enable
tr, tf
CB1
tCB2
tRS2*
CB2

*Assumes part was deselected during any previous E pulse.

FIGURE 13 — INTERRUPT PULSE WIDTH AND $\overline{\text{IRQ}}$ RESPONSE


PWI
CA1, 2
CB1, 2
IRQA/B
tRS3*

*Assumes Interrupt Enable Bits are set.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 14 — $\overline{\text{IRQ}}$ RELEASE TIME

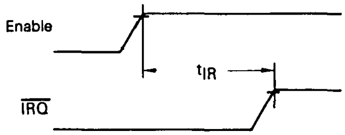
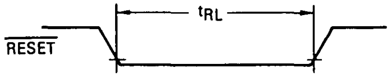

Enable
t_{IR}
$\overline{\text{IRQ}}$

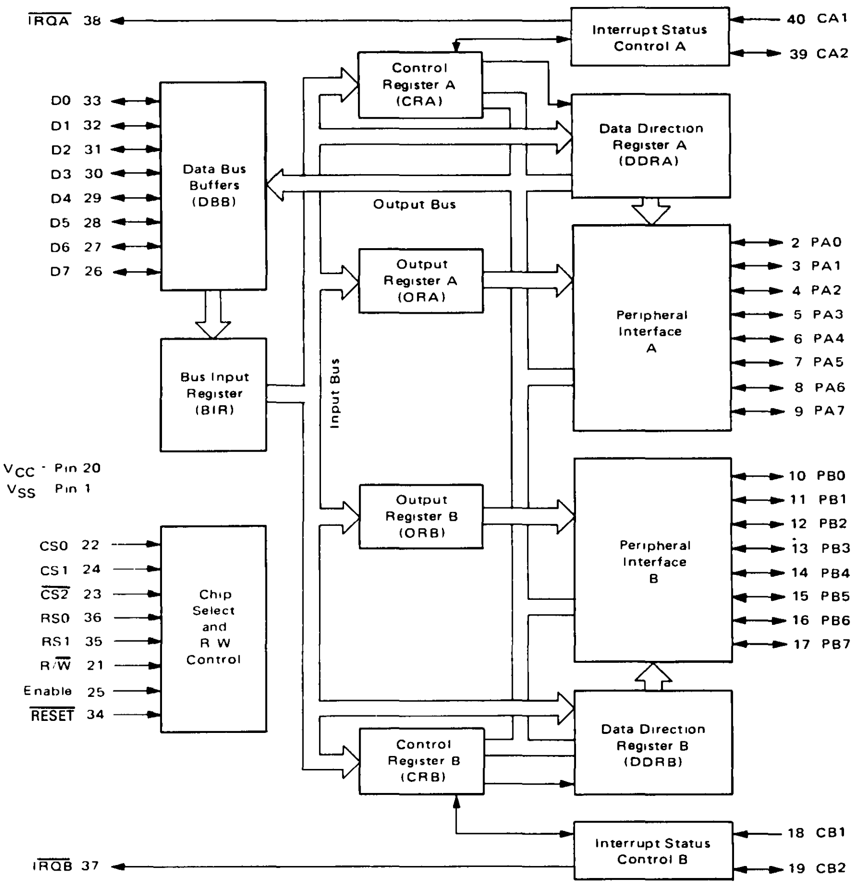
FIGURE 15 — $\overline{\text{RESET}}$ LOW TIME


t_{RL}
$\overline{\text{RESET}}$

*The $\overline{\text{RESET}}$ line must be a V_{IH} for a minimum of 1.0 µs before addressing the PIA

Note: Timing measurements are referenced to and from a low voltage of 0 8 volts and a high voltage of 2 0 volts, unless otherwise noted.

FIGURE 16 — EXPANDED BLOCK DIAGRAM


$\overline{\text{IRQA}}$ 38
Interrupt Status Control A
40 CA1
39 CA2
Control Register A (CRA)
Data Direction Register A (DDRA)
D0 33
D1 32
D2 31
D3 30
D4 29
D5 28
D6 27
D7 26
Data Bus Buffers (DBB)
Output Bus
Output Register A (ORA)
Peripheral Interface A
2 PA0
3 PA1
4 PA2
5 PA3
6 PA4
7 PA5
8 PA6
9 PA7
Bus Input Register (BIR)
Input Bus
V_{CC} - Pin 20
V_{SS} Pin 1
Output Register B (ORB)
Peripheral Interface B
10 PB0
11 PB1
12 PB2
13 PB3
14 PB4
15 PB5
16 PB6
17 PB7
CS0 22
CS1 24
$\overline{\text{CS2}}$ 23
RS0 36
RS1 35
R/$\overline{\text{W}}$ 21
Enable 25
$\overline{\text{RESET}}$ 34
Chip Select and R W Control
Data Direction Register B (DDRB)
Control Register B (CRB)
Interrupt Status Control B
18 CB1
19 CB2
$\overline{\text{IRQB}}$ 37

PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the M6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802, or MC6808 microprocessors, VMA should be used as an active part of the address decoding

Bidirectional Data (D0-D7) – The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation The read/write line is in the read (high) state when the PIA is selected for a read operation.

Enable (E) – The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

Read/Write (R/$\overline{\text{W}}$) – This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus The PIA output buffers are enabled when the proper address and the enable pulse E are present.

$\overline{\text{RESET}}$ – The active low $\overline{\text{RESET}}$ line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and $\overline{\text{CS2}}$) – These three input signals are used to select the PIA. CS0 and CS1 must be high and $\overline{\text{CS2}}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) – The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) – The active low Interrupt Request lines ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bidirectional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) – Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low." Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PB0-PB7) – The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7. They have three-state capabiity, allowing them to enter a high-impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines

PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of at least 1 milliampere at 1.5 volts to directly drive the base of a transistor switch

Interrupt Input (CA1 and CB1) — Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) — The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL, as an input the internal pullup resistor on this line represents 1.5 standard TTL loads. The function of this signal line is programmed with Control Register A

Peripheral Control (CB2) — Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 – INTERNAL ADDRESSING

RS1	RS0	Control Register Bit CRA-2	 CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 17, the MC6821 has a pair of I/O ports whose characteristics differ greatly. The A side is designed to drive CMOS logic to normal 30% to 70% levels, and incorporates an internal pullup device that remains connected even in the input mode. Because of this, the A side requires more drive current in the input mode than Port B. In contrast, the B side uses a normal three-state NMOS buffer which cannot pullup to CMOS levels without external resistors. The B side can drive extra loads such as Darlingtons without problem. When the PIA comes out of reset, the A port represents inputs with pullup resistors, whereas the B side (input mode also) will float high or low, depending upon the load connected to it.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

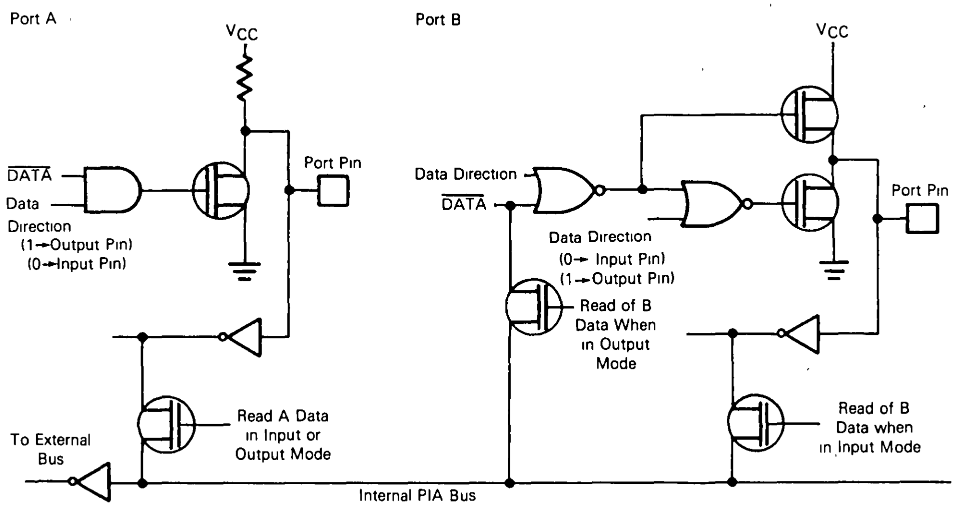
Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) — The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

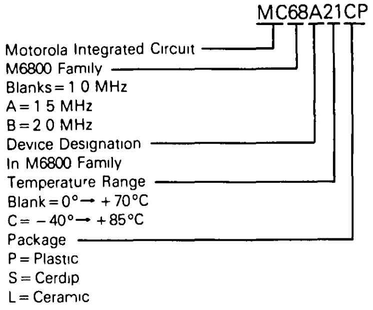
Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) — Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1) When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-0, CRA-1, and CRB-1) — The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 17 — PORT A AND PORT B EQUIVALENT CIRCUITS


Port A
V_{CC}
$\overline{\text{DATA}}$
Data Direction (1→Output Pin) (0→Input Pin)
Port Pin
Read A Data in Input or Output Mode
To External Bus
Internal PIA Bus
Port B
V_{CC}
Data Direction
$\overline{\text{DATA}}$
Data Direction (0→ Input Pin) (1→Output Pin)
Port Pin
Read of B Data When in Output Mode
Read of B Data when in Input Mode

ORDERING INFORMATION


MC68A21CP

Motorola Integrated Circuit
M6800 Family
Blanks = 1.0 MHz
A = 1.5 MHz
B = 2.0 MHz
Device Designation
In M6800 Family
Temperature Range
Blank = 0° → +70°C
C = −40° → +85°C
Package
P = Plastic
S = Cerdip
L = Ceramic

BETTER PROGRAM

Better program processing is available on all types listed. Add suffix letters to part number.

Level 1 add "S" Level 2 add "D" Level 3 add "DS"

Level 1 "S" = 10 Temp Cycles — (−25 to 150°C), Hi Temp testing at T_A max
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2

FIGURE 18 — CONTROL WORD FORMAT

Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 — (bit 7)

b1 = 0. IRQA(B)1 set by high-to-low transition on CA1 (CB1)

b1 = 1 IRQA(B)1 set by low-to-high transition on CA1 (CB1)

CA1 (CB1) Interrupt Request Enable/Disable

b0 = 0 Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition [1]

b0 = 1 Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition

[1] IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled

IRQA(B) 1 Interrupt Flag (bit 7)

Goes high on active transition of CA1 (CB1), Automatically cleared by MPU Read of Output Register A(B) May also be cleared by hardware Reset

Control Register

b7	b6	b5	b4	b3	b2	b1	b0
IRQA(B)1 Flag	IRQA(B)2 Flag	CA2 (CB2) Control			DDR Access	CA1 (CB1) Control	

IRQA(B)2 Interrupt Flag (bit 6)

When CA2 (CB2) is an input, IRQA(B) goes high on active transition CA2 (CB2), Automatically cleared by MPU Read of Output Register A(B) May also be cleared by hardware Reset

CA2 (CB2) Established as Output (b5 = 1) IRQA(B) 2 = 0, not affected by CA2 (CB2) transitions

Determines Whether Data Direction Register Or Output Register is Addressed

b2 = 0 Data Direction Register selected

b2 = 1 Output Register selected

CA2 (CB2) Established as Output by b5 = 1

(Note that operation of CA2 and CB2 output functions are not identical)

b5 b4 b3

1 0 → CA2

b3 = 0 **Read Strobe with CA1 Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A, returned high by next active CA1 transition, as specified by bit 1

b3 = 1 **Read Strobe with E Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A, returned high by next high-to-low E transition during a deselect

→ CB2

b3 = 0 **Write Strobe with CB1 Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B, returned high by the next active CB1 transition as specified by bit 1 CRB-b7 must first be cleared by a read of data

b3 = 1 **Write Strobe with E Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B, returned high by the next low-to-high E transition following an E pulse which occurred while the part was deselected

b5 b4 b3

1 1 → **Set/Reset CA2 (CB2)**

CA2 (CB2) goes low as MPU writes b3 = 0 into Control Register

CA2 (CB2) goes high as MPU writes b3 = 1 into Control Register

CA2 (CB2) Established as Input by b5 = 0

b5 b4 b3

0 → **CA2 (CB2) Interrupt Request Enable/Disable**

b3 = 0 Disables IRQA(B) MPU Interrupt by CA2 (CB2) active transition *

b3 = 1 Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition

*IRQA(B) will occur on next (MPU generated) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled

→ **Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 — (Bit b6)**

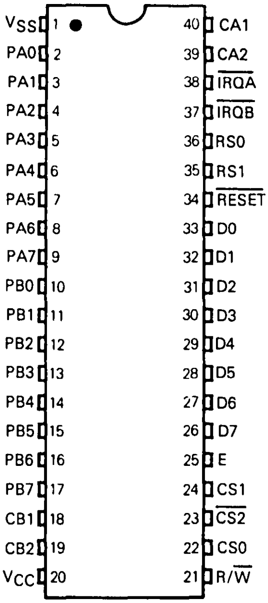
b4 = 0 IRQA(B)2 set by high-to-low transition on CA2 (CB2)

b4 = 1 IRQA(B)2 set by low-to-high transition on CA2 (CB2)

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip S Suffix	1.0	0°C to 70°C	MC6821S
	1.0	−40°C to 85°C	MC6821CS
	1.5	0°C to 70°C	MC68A21S
	1.5	−40°C to 85°C	MC68A21CS
	2.0	0°C to 70°C	MC68B21S
Plastic P Suffix	1.0	0°C to 70°C	MC6821P
	1.0	−40°C to 85°C	MC6821CP
	1.5	0°C to 70°C	MC68A21P
	1.5	−40°C to 85°C	MC68A21CP
	2.0	0°C to 70°C	MC68B21P

PIN ASSIGNMENT


V_{SS} 1
PA0 2
PA1 3
PA2 4
PA3 5
PA4 6
PA5 7
PA6 8
PA7 9
PB0 10
PB1 11
PB2 12
PB3 13
PB4 14
PB5 15
PB6 16
PB7 17
CB1 18
CB2 19
V_{CC} 20
40 CA1
39 CA2
38 $\overline{IRQA}$
37 $\overline{IRQB}$
36 RS0
35 RS1
34 $\overline{RESET}$
33 D0
32 D1
31 D2
30 D3
29 D4
28 D5
27 D6
26 D7
25 E
24 CS1
23 $\overline{CS2}$
22 CS0
21 R/$\overline{W}$

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC146823

Advance Information

CMOS Parallel Interface

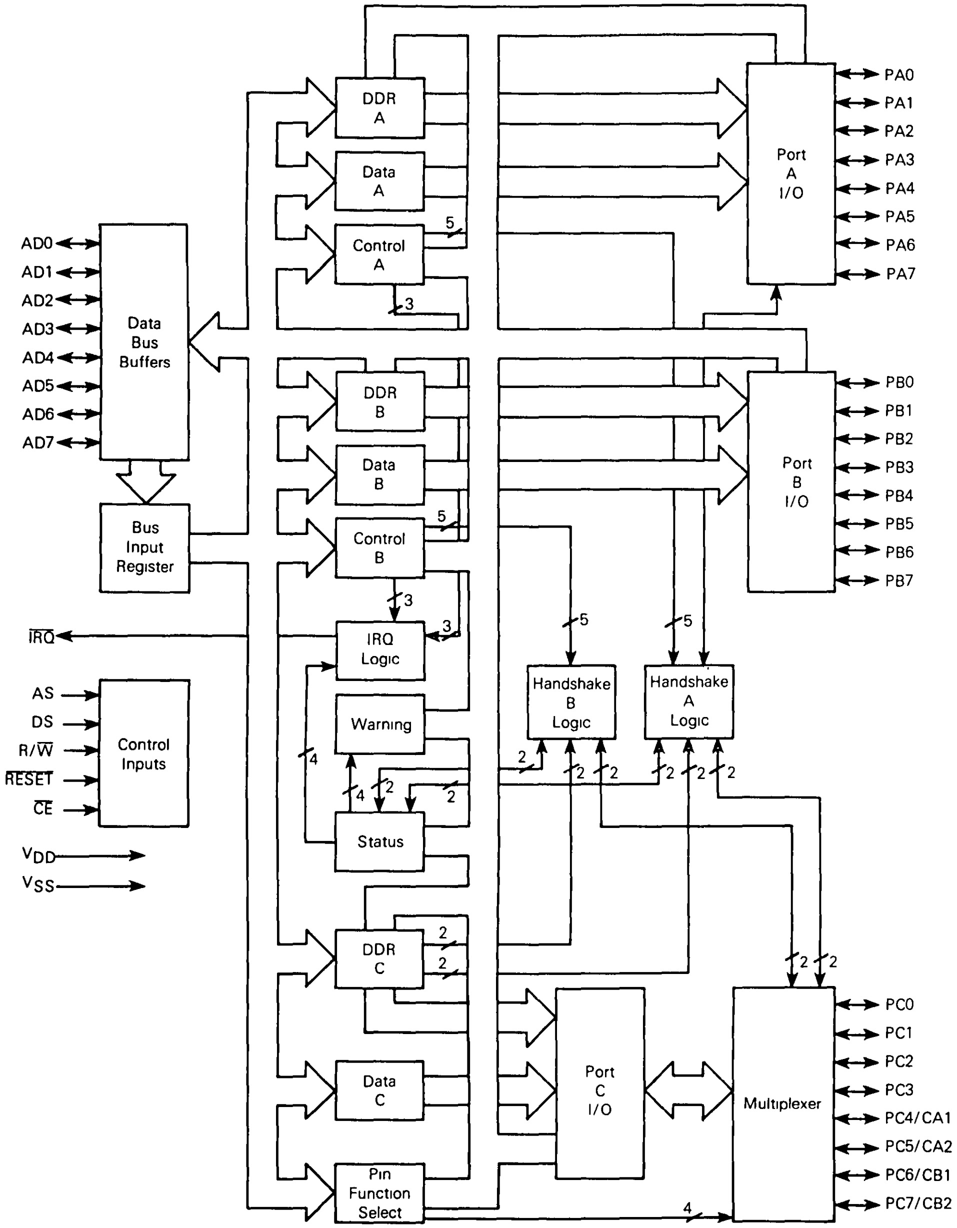
The MC146823 CMOS parallel interface (CPI) provides a universal means of interfacing external signals with the MC146805E2 CMOS microprocessor and other multiplexed bus microprocessors. The unique MOTEL circuit on-chip allows direct interfacing to most industry CMOS microprocessors, as well as many NMOS MPUs.

The MC146823 CPI includes three bidirectional 8-bit ports or 24 I O pins. Each I O line may be separately established as an input or an output under program control via data direction registers associated with each port. Using the bit change and test instructions of the MC146805E2, each individual I O pin can be separately accessed. All port registers are read write bytes to accommodate read-modify-write instructions. Features include.

- 24 Individually Programmed I O Pins
- MOTEL Circuit for Bus Compatibility with Many Microprocessors
- Multiplexed Bus Compatibility with: MC146805E2, MC6801, MC6803, and Competitive Microprocessors
- Data Direction Registers for Ports A, B, and C
- Four Port C I O Pins May Be Used as Control Lines for
    - Four Interrupt Inputs
    - Input Byte Latch
    - Output Pulse
    - Handshake Activity
- 15 Registers Addressed as Memory Locations
- Handshake Control Logic for Input and Output Peripheral Operation
- Interrupt Output Pin
- Reset Input to Clear Interrupts and Initialize Internal Registers
- 3.0 Volt to 5.5 Volt Operating V_{DD}

This document contains information on a new product. Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM


AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
Data Bus Buffers
Bus Input Register
IRQ
AS
DS
R/W
RESET
CE
Control Inputs
VDD
VSS
DDR A
Data A
Control A
DDR B
Data B
Control B
IRQ Logic
Warning
Status
DDR C
Data C
Pin Function Select
Port A I/O
Port B I/O
Port C I/O
Handshake B Logic
Handshake A Logic
Multiplexer
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4/CA1
PC5/CA2
PC6/CB1
PC7/CB2
5
3
5
3
3
4
4
2
2
5
5
2
2
2
2
2
2
2
2
2
2
2
4

MAXIMUM RATINGS (Voltages reference to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages	V_{in}	V_{SS}−0.5 to V_{DD}+0.5	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

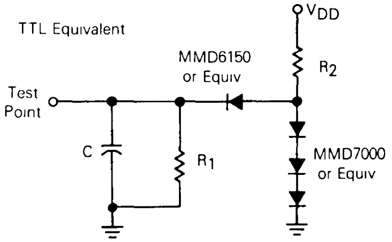
Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic PLCC	θ_{JA}	 100 100	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \geq (V_{in}$ or $V_{out}) \geq V_{DD}$. Leakage currents are reduced and reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD}=5 Vdc ±10%, V_{SS}=0 Vdc, T_A=0°C to 70°C, unless otherwise noted)

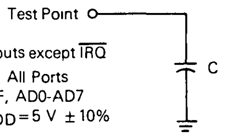
Parameter	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10$ μA)	V_{OL} V_{OH}	– V_{DD}−0.1	0.1 –	V V
Output High Voltage (I_{Load}= −1.6 mA) AD0-AD7 (I_{Load}= −0.2 mA) PA0-PA7, PC0-PC7 (I_{Load}= −0.36 mA) PB0-PB7	 V_{OH} V_{OH} V_{OH}	 4.1 4.1 4.1	 V_{DD} V_{DD} V_{DD}	V
Output Low Voltage (I_{Load}=1.6 mA) AD0-AD7, PB0-PB7 (I_{Load}=0.8 mA) PA0-PA7, PC0-PC7 (I_{Load}=1.0 mA) $\overline{IRQ}$	 V_{OL} V_{OL} V_{OL}	 V_{SS} V_{SS} V_{SS}	 0.4 0.4 0.4	V
Input High Voltage, AD0-AD7, AS, DS, R/$\overline{W}$, $\overline{CE}$, PA0-PA7, PB0-PB7, PC0-PC7 $\overline{RESET}$	V_{IH} V_{IH}	V_{DD}−2.0 V_{DD}−0.8	V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Quiescent Current — No dc Loads (All Ports Programmed as Inputs, All Inputs=V_{DD} − 0.2 V)	I_{DD}	–	160	μA
Total Supply Current (All Ports Programmed as Inputs, CE=V_{IL}, t_{cyc}=1 μs)	I_{DD}	–	3.0	mA
Input Current, $\overline{CE}$, AS, R/$\overline{W}$, DS, $\overline{RESET}$	I_{in}	–	±1.0	μA
Hi-Z State Leakage, AD0-AD7, PA0-PA7, PB0-PB7, PC0-PC7	I_{TSL}	–	±10.0	μA

EQUIVALENT TEST LOADS

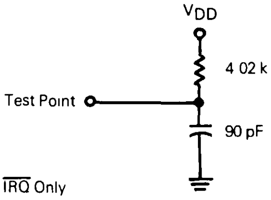


Pin	R1	R2	C
AD0-AD7	2.55k	2.0k	130 pF
PA0-PA7, PC0-PC7	20.0k	4.32k	50 pF
PB0-PB7	11.5k	2.1k	50 pF

CMOS Equivalent



For all outputs except $\overline{IRQ}$
C=50 pF, All Ports
=130 pF, AD0-AD7
for V_{DD}=5 V ±10%



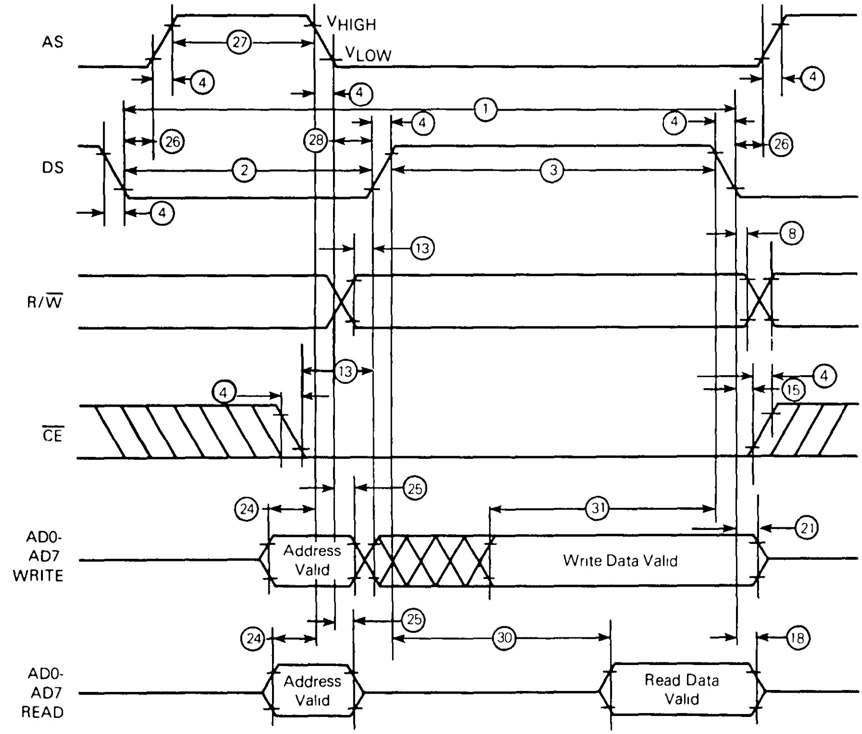
$\overline{IRQ}$ Only

BUS TIMING (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Ident. Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1000	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High	PW_{EL}	300	–	ns
3	Pulse Width, DS/E High or $\overline{RD}$/$\overline{WR}$ Low	PW_{EH}	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	ns
13	R/$\overline{W}$ and $\overline{CE}$ Setup Time Before DS/E	t_{RWS}	25	–	ns
15	Chip Enable Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	100	ns
21	Write Data Hold Time	t_{DHW}	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	25	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	60	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	170	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	220	–	ns

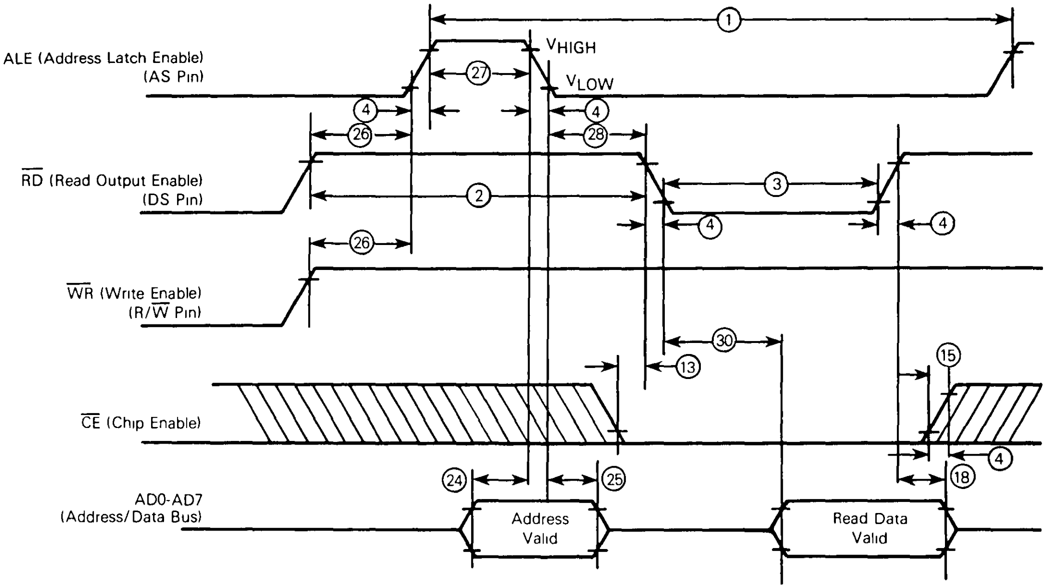
NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals

BUS TIMING DIAGRAM

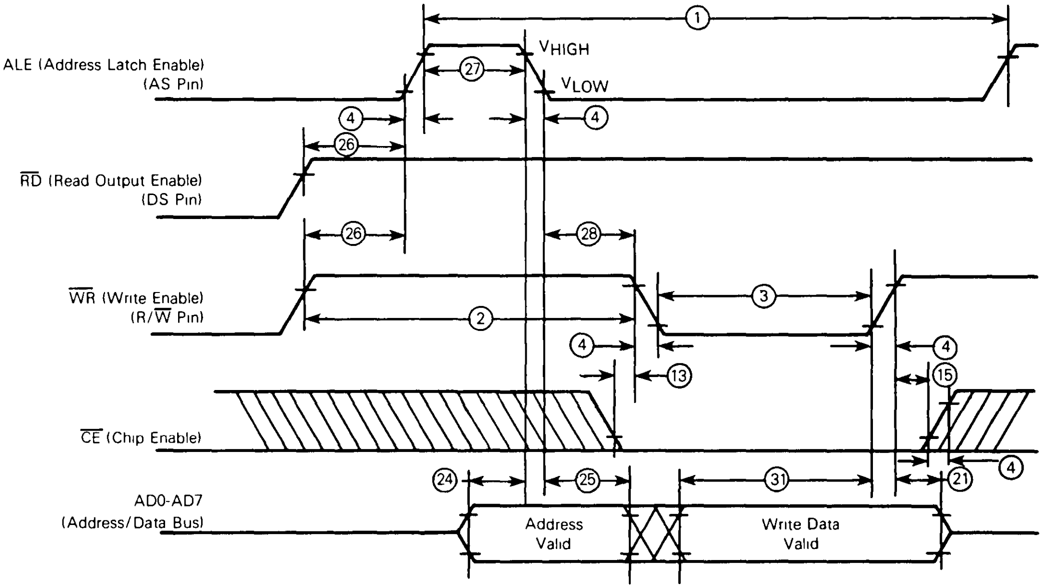


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

BUS READ TIMING COMPETITOR MULTIPLEXED BUS


ALE (Address Latch Enable) (AS Pin)
RD (Read Output Enable) (DS Pin)
WR (Write Enable) (R/W Pin)
CE (Chip Enable)
AD0-AD7 (Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑬
⑮
⑱
㉔
㉕
㉖
㉗
㉘
㉚
Address Valid
Read Data Valid

BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS


ALE (Address Latch Enable) (AS Pin)
RD (Read Output Enable) (DS Pin)
WR (Write Enable) (R/W Pin)
CE (Chip Enable)
AD0-AD7 (Address/Data Bus)
VHIGH
VLOW
①
②
③
④
⑬
⑮
㉑
㉔
㉕
㉖
㉗
㉘
㉛
Address Valid
Write Data Valid

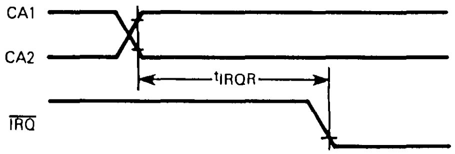
NOTE $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C)

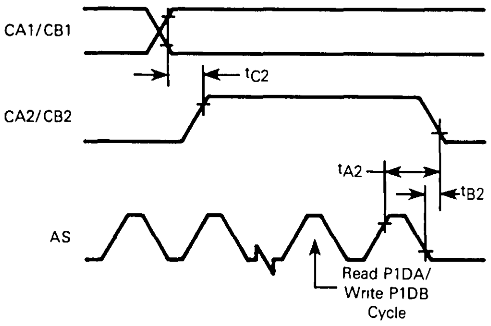
Parameter	Symbol	Min	Max	Unit
Interrupt Response (Input Modes 1 and 3)	t_{IRQR}	—	1 0	µs
Delay, CA1 (CB1) Active Transition to CA2 (CB2) High (Output Mode 0)	t_{C2}	—	1 0	µs
Delay, CA2 Transition from Positive Edge of AS (Output Modes 0 and 1)	t_{A2}	—	1 0	µs
Delay, CB2 Transition from Negative Edge of AS (Output Modes 0 and 1)	t_{B2}	—	1 0	µs
CA2/CB2 Pulse Width (Output Mode 1)	t_{PW}	0 5	1 5	ns
Delay, V_{DD} Rise to $\overline{\text{RESET}}$ High	t_{RLH}	1 0	—	µs
Pulse Width, $\overline{\text{RESET}}$	t_{RW}	1 0	—	ns

CONTROL TIMING DIAGRAMS

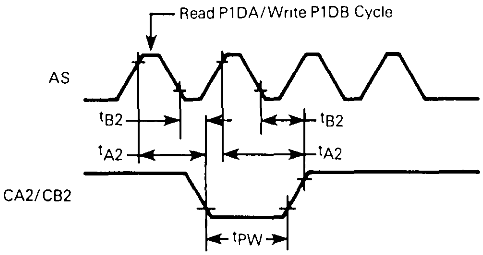
$\overline{\text{IRQ}}$ RESPONSE (INPUT MODES 1 AND 3)


CA1
CA2
t_{IRQR}
$\overline{\text{IRQ}}$

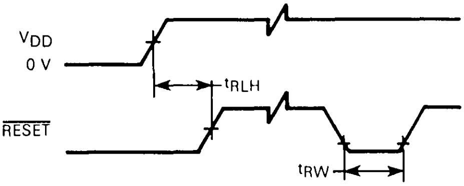
CA2/CB2 DELAY (OUTPUT MODE 0)


CA1/CB1
t_{C2}
CA2/CB2
t_{A2}
t_{B2}
AS
Read P1DA/
Write P1DB
Cycle

CA2/CB2 DELAY (OUTPUT MODE 1)


Read P1DA/Write P1DB Cycle
AS
t_{B2}
t_{B2}
t_{A2}
t_{A2}
CA2/CB2
t_{PW}

$\overline{\text{RESET}}$


V_{DD}
0 V
t_{RLH}
$\overline{\text{RESET}}$
t_{RW}

GENERAL DESCRIPTION

The MC146823, CMOS parallel interface (CPI), contains 24 individual bidirectional I/O lines configured in three 8-bit ports. The 15 internal registers, which control the mode of operation and contain the status of the port pins, are accessed via an 8-bit multiplexed address/data bus. The lower four address bits (AD0-AD3) of the multiplexed address bus determine which register is to be accessed (see Figure 1). The four address bits (AD4, AD5, AD6, and AD7) must be separately decoded to position this memory map within each 256 byte address space available via the 8-bit multiplexed address bus For more detailed information refer to **REGISTER DESCRIPTION.**

FIGURE 1 — REGISTER ADDRESS MAP

0	Port A Data, Clear CA1 Interrupt	P1DA
1	Port A Data, Clear CA2 Interrupt	P2DA
2	Port A Data	PDA
3	Port B Data	PDB
4	Port C Data	PDC
5	Not Used	–
6	Data Direction Register for Port A	DDRA
7	Data Direction Register for Port B	DDRB
8	Data Direction Register for Port C	DDRC
9	Control Register for Port A	CRA
A	Control Register for Port B	CRB
B	Pin Function Select Register for Port C	FSR
C	Port B Data, Clear CB1 Interrupt	P1DB
D	Port B Data, Clear CB2 Interrupt	P2DB
E	Handshake/Interrupt Status Register	HSR
F	Handshake Over-Run Warning Register	HWR

The CPI is implemented with the MOTEL circuit which allows direct interface with either of the two major multiplexed microprocessor bus types A detailed description of the MOTEL circuit is provided in the **MOTEL** section.

Three data direction registers (DDRs), one for each port, determine which pins are outputs and which are inputs. A logic zero on a DDR bit configures its associated pin as an input; and a logic one configures the pin as an output. Upon reset, the DDRs are cleared to logic zero to configure all port pins as inputs.

Actual port data may be read or written via the port data registers (PDA, PDB, and PDC). Ports A and B each have two additional data registers (P1DA and P2DA — P1DB and P2DB) which are used to clear the associated handshake/interrupt status register bits (HSA1 and HSA2 — HSB1 and HSB2), respectively. Port A may also be configured as an 8-bit latch when used with CA1. Reset has no effect on the contents of the port data registers. Users are advised to initialize the port data registers before changing any port pin to an output.

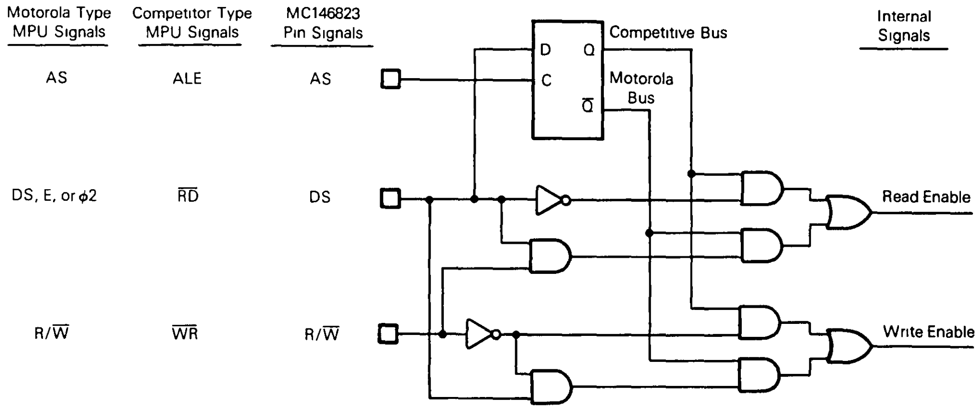
Four pins on port C (PC4/CA1, PC5/CA2, PC6/CB1, and PC7/CB2) may additionally be programmed as handshake lines for ports A and B via the port C function select register (FSR). Both ports A and B have one input-only line and one bidirectional handshake line each associated with them The handshake lines may be programmed to perform a variety of tasks such as interrupt requests, setting flags, latching data, and data transfer requests and/or acknowledgements. The handshake functions are programmed via control registers A and B (CRA and CRB). Additional information may be found in **PIN DESCRIPTIONS, REGISTER DESCRIPTION,** or **HANDSHAKE OPERATION.**

MOTEL

The MOTEL circuit is a concept that permits the MC146823 to be directly interfaced with different types of multiplexed bus microprocessors without any additional external logic. For a more detailed description of the multiplexed bus, see **MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7).** Most multiplexed microprocessors use one of two synchronous buses to interface peripherals. One bus was originated by Motorola in the MC6803 and the other by Intel in the 8085.

The MOTEL circuit (for MOTorola and intEL bus) is built into peripheral and memory ICs to permit direct connection to either type of bus. A functional diagram of the MOTEL circuit is shown in Figure 2.

FIGURE 2 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



The microprocessor type is automatically selected by the MOTEL circuit through latching the state of the DS/$\overline{RD}$ pin with AS/ALE. Since DS is always low during AS and $\overline{RD}$ is always high during ALE, the latch automatically indicates with which type microprocessor bus it is interfaced.

PIN DESCRIPTIONS

The following paragraphs contain a brief description of the input and output pins. References (if applicable) are given to other paragraphs that contain more detail about the function being performed.

MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7)

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion of the bus cycle for data. Address-then-data multiplexing does not slow the access time of the MC146823 since the bus reversal from address to data is occurring during the internal register access time.

The address must be valid t_{ASL} prior to the fall of AS/ALE at which time the MC146823 latches the address present on the AD0-AD3 pins. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146823 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to high impedance) t_{DHR} hold time after DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case

ADDRESS STROBE (AS)

The address strobe input pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the addresses AD0-AD3 to be latched within the MC146823. The automatic MOTEL circuit in the MC146823 also latches the state of the DS pin with the falling edge of AS or ALE

DATA STROBE OR READ (DS)

The DS input pin has two interpretations via the MOTEL circuit. When generated by a Motorola microprocessor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), or phase 2 (phase 2 clock). During read cycles, DS or $\overline{RD}$ signifies the time that the CPI is to drive the bidirectional bus In write cycles, the trailing edge of DS or rising edge of $\overline{WR}$ causes the parallel interface to latch the written data present on the bidirectional bus.

The second MOTEL interpretation of DS is that of $\overline{RD}$, MEMR, or I/OR originating from the competitor's microprocessor. In this case, DS identifies the time period when the parallel interface drives the bus with read data This interpretation of DS is also the same as an output-enable signal on a typical memory

The MOTEL circuit, within the MC146823, latches the state of the DS pin on the falling edge of AS/ALE When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus microprocessors. To insure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

READ/WRITE (R/$\overline{W}$)

The MOTEL circuit treats the R/$\overline{W}$ input pin in one of two ways First, when a Motorola microprocessor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ while DS is high

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and I/$\overline{OW}$ from competitor's microprocessors The MOTEL circuit in this mode gives the R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs

CHIP ENABLE ($\overline{CE}$)

The $\overline{CE}$ input signal must be asserted (low) for the bus cycle in which the MC146823 is to be accessed $\overline{CE}$ is not latched and must be stable prior to and during DS (in the Motorola case of MOTEL) and prior to and during $\overline{RD}$ and $\overline{WR}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the MC146823. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state

When $\overline{CE}$ is high, all data, DS, and R/$\overline{W}$ inputs from the microprocessor are disconnected within the MC146823 This permits the MC146823 to be isolated from a powered-down microprocessor

RESET ($\overline{RESET}$)

The $\overline{RESET}$ input pin is an active-low line that is used to restore all register bits, except the port data register bits, to logical zeros After reset, all port lines are configured as inputs and no interrupt or handshake lines are enabled

INTERRUPT REQUEST ($\overline{IRQ}$)

The $\overline{IRQ}$ output line is an open drain active-low signal that may be used to interrupt the microprocessor with a service request. The "open-drain" output allows this and other interrupt request lines to be wire ORed with a pullup resistor The $\overline{IRQ}$ line is low when bit 7 of the status register is high Bit 7 (IRQF) of the handshake/interrupt status register (HSR) is set if any enabled handshake transition occurs, and its associated control register bit is set to allow interrupts Refer to **INTERRUPT DESCRIPTION** or **HANDSHAKE OPERATION** for additional information

PORT A, BIDIRECTIONAL I/O LINES (PA0-PA7)

Each line of port A, PA0-PA7, is individually programmable as either an input or output via its data direction register (DDRA) An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one See Figure 3 for typical I/O circuitry and Table 1 for I/O operation

There are three data registers associated with port A PDA, P1DA, and P2DA. P1DA and P2DA are accessed when certain handshake activity is desired See **HANDSHAKE OPERATION** for more information

Data written to the port A data register, PDA, is latched into the port A output latch regardless of the state of the DDRA. Data written to P1DA or P2DA is ignored and has no affect upon the output data latch or the I/O lines An MPU read of port bits programmed as outputs reflect the last value written to the PDA register Port A pins programmed as inputs may be latched via the handshake line PC4/CA1 (see

FIGURE 3 — TYPICAL PORT I/O CIRCUITRY

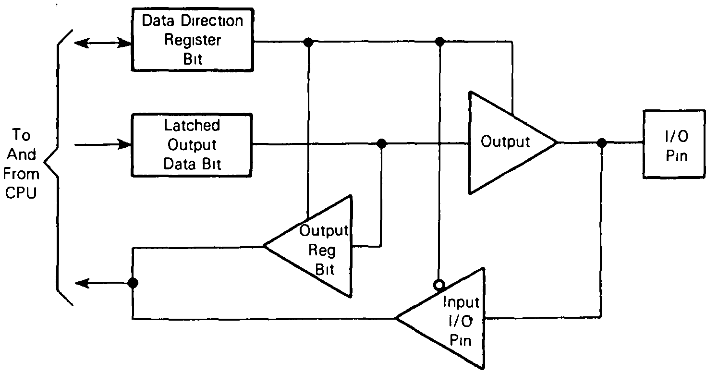


TABLE 1 — PORT DATA REGISTER ACCESSES (ALL PORTS)

R/$\overline{W}$	DDR Bit	Results
0	0	The I/O pin is in input mode Data is written into the output data latch
0	1	Data is written into the output data latch and output to the I/O pin
1	0	The state of the I/O pin is read
1	1	The I/O pin is in an output mode The output data latch is read

HANDSHAKE OPERATION) and latched input data may be read via any of the three port A data registers. If the port A input latch feature is not enabled, an MPU read of any port A data register reflects the current status of the port A input pins if the corresponding DDRA bits equal zero Reset has no effect upon the contents of the port A data register; however, all pins will be placed in the input mode (all DDRA bits forced to equal zero) and all handshake lines will be disabled.

PORT B BIDIRECTIONAL I/O LINES (PB0-PB7)

Each line of port B, PB0-PB7, is individually programmable as either an input or an output via its data direction register (DDRB) An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one.

There are three data registers associated with port B: PDB, P1DB, and P2DB. PDB is used for simple port B data reads and writes. P1DB and P2DB are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information

Data written to PDB or P1DB data register is latched into the port B output latch regardless of the state of the DDRB An MPU read of port bits programmed as outputs reflect the last value written to a port B data register An MPU read of any port B register reflects the current status of the input pins whose DDRB bits equal zero Reset has no effect upon the contents of the port B data register; however, all pins will be placed in the input mode (all DDRB bits forced to equal zero) and all handshake lines will be disabled.

PORT C, BIDIRECTIONAL I/O LINES (PC0-PC3)

Each line of port C, PC0-PC3, is individually programmable as either an input or an output via its data direction register (DDRC). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. Port C data register (PDC) is used for simple port C data reads and writes

Data written into PDC is latched into the port C data latch regardless of the state of the DDRC An MPU read of port C bits programmed as outputs reflect the last value written to the PDC register. An MPU read of the port C register reflects the current status of the corresponding input pins whose DDRC bits equal zero. Reset has no effect upon the contents of the port C data register; however, all pins will be placed in the input mode (all DDRC bits forced to equal zero) and all handshake lines will be disabled.

PORT C BIDIRECTIONAL I/O LINE OR PORT A INPUT HANDSHAKE LINE (PC4/CA1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR) If programmed as a port C I/O pin, PC4/CA1 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC4/CA1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT A BIDIRECTIONAL HANDSHAKE LINE (PC5/CA2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC5/CA2 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC5/CA2 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B INPUT HANDSHAKE LINE (PC6/CB1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O pin, PC6/CB1 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC6/CB1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B BIDIRECTIONAL HANDSHAKE LINE (PC7/CB2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR) If programmed as a port C I/O line, PC7/CB2 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC7/CB2 performs as described in **HANDSHAKE OPERATION.**

HANDSHAKE OPERATION

Up to four port C pins can be configured as handshake lines for ports A and B (one input-only and one bidirectional line for each port) via the port C function select register (FSR). The direction of data flow for the two bidirectional handshake lines (CA2 and CB2) is determined by bits 5 and 7, respectively, of the port C data direction register (DDRC). Actual handshake operation is defined by the appropriate port control register (CRA or CRB).

The control registers allow each handshake line to be programmed to operate in one of four modes. CA2 and CB2 each have four input and four output modes For detailed information, see Tables 2 and 3.

A summary of the handshake modes is given in the input and output sections that follow All handshake activity is disabled by reset.

INPUT

Handshake lines programmed as inputs operate in any of four different modes as defined by the control registers (see Table 2). A bit in the handshake/interrupt status register (HSR) is set to a logic one on an active transition of any handshake line programmed as an input Modes 0 and 1 define a negative transition as active, modes 2 and 3 define a positive transition as active If modes 1 or 3 are selected on any input handshake line then the active transition of that line results in the IRQF bit of the HSR being set to a logic one and causes the interrupt line ($\overline{IRQ}$) to go low $\overline{IRQ}$ is released by clearing the HSR bits that are input handshake lines which have interrupts enabled

If an active transition occurs while the associated HSR bit is set to a logic one, the corresponding bit in the handshake warning register (HWR) is set to a logic one indicating that service of at least one active transition was missed An HWR bit is cleared to a logic zero by first accessing the appropriate port data register, to clear the appropriate HSR status bit, followed by a read of the HWR

TABLE 2 — INPUT HANDSHAKE MODES

Mode	Control Register Bits*	Active Edge	Status Bit In HSR	$\overline{IRQ}$ Pin
0	00	– Edge	Set high on active edge	Disabled
1	01	– Edge	Set high on active edge	Goes low when corresponding status flag in HSR goes high
2	10	+ Edge	Set high on active edge	Disabled
3	11	+ Edge	Set high on active edge	Goes low when corresponding status flag in HSR goes high

*Cleared to logic zero on reset

TABLE 3 — OUTPUT HANDSHAKE LINES (CA2 AND CB2 ONLY)

Mode	Control Register CRA(B) Bits 3 and 4*	Handshake Line Set High	Handshake Line Cleared Low	Default Level
0	00	Handshake set high on active transition of CA1 input. Handshake set high on active transition of CB1 input	Read of P1DA or a read of P2DA while HSA1 is cleared. Write of port B P1DB or write of P2DB while HSB1 is cleared.	High
1	01	High on the first positive (negative) transition of AS while CA2 (CB2) is low.	Low on the first positive (negative) transition on AS following a read (write) of port A(B) data registers P1DA(B) or P2DA(B).	High
2	10	Never	Always	Low
3	11	Always	Never	High

*Cleared to logic zero on reset.

INPUT LATCH

Port A input-only handshake line (PC4/CA1) can be programmed to function as a latch enable for port A input data via CA1 LE (bit 2 of CRA) If CA1 LE is programmed to a logic one, an active transition of PC4/CA1 will latch the current status of the port A input pins into all three port A data registers (PDA, P1DA, and P2DA) When CA1 LE is enabled, port A and PC4/CA1 function as an 8-bit transparent latch, that is, if the HSA1 bit in the HSR is a logic zero then a read of any port A register reflects the current state of the port A input pins and corresponding bits of the output data latch for port A output pins. If HSA1 is a logic one, a read of any port A data register reflects the state of the port A input pins when HSA1 was set and the corresponding bits of the port A output data latch for port A output pins

Further transitions of PC4/CA1 result only in setting the HWA1 bit in the HWR and do not relatch data into the port A registers Latched data is released only by clearing HSA1 in the HSR to a logic zero (HSA1 is cleared by reading P1DA).

OUTPUT

Each bidirectional handshake line programmed as an output by the DDRC operates in one of four modes as described in Table 3 Modes 2 and 3 force the output handshake line to reflect the state of bit 4 in the appropriate control register

In modes 0 and 1, PC5/CA2 is forced low during the cycle following a read of P1DA or a read of P2DA while HSA1 is cleared. PC7/CB2 is forced low during the cycle following a write to P1DB or a write to P2DB while HSB1 is cleared. Because of these differences, port A is the preferred input port and port B is the preferred output port.

In mode 0, PC5/CA2 (PC7/CB2) is set high by an active transition of PC4/CA1 (PC6/CB1). In mode 1, PC5/CA2 (PC7/CB2) is set high in the cycle following the cycle in which PC5/CA2 (PC7/CB2) goes low Mode 1 forces a low-going pulse on PC5/CA2 (PC7/CB2) following a read (write) of P1DA (P1DB) or P2DA (P2DB) that is approximately one cycle time wide

When entering an output handshake mode for the first time after a reset, the handshake line outputs the default level as listed in Table 3

INTERRUPT DESCRIPTION

The MC146823 allows an MPU interrupt request ($\overline{\text{IRQ}}$ low) via the input handshake lines The input handshake line, operating in modes 1 or 3 as defined by the control registers (CRA and CRB), causes $\overline{\text{IRQ}}$ to go low when IRQF (interrupt flag) in the HSR is set to a logic one. $\overline{\text{IRQ}}$ is released when IRQF is cleared See **Handshake/Interrupt Status Register** under **REGISTER DESCRIPTION** for additional information.

REGISTER DESCRIPTION

The MC146823 has 15 registers (see Figure 1) which define the mode of operation and status of the port pins The following paragraphs describe these registers

Register Names:

Control Register A (CRA)
Control Register B (CRB)

Register Addresses:

$9 (CRA)
$A (CRB)

Register Bits:

	7	6	5	4	3	2	1	0
$9	X	X	X	CA2 Mode		CA1 LE	CA1 Mode	
$A	X	X	X	CB2 Mode		X	CB1 Mode	

Purpose:

These two registers control the handshake and interrupt activity for those pins defined as handshake lines by the port C function select register (FSR)

Description:

CA2 and CB2 are programmed as inputs or outputs via the associated DDRC bits Each handshake line is controlled by two mode bits Bit 2 of CRA enables the Port A latch for an active CA1 transition. Table 2 describes the input handshake modes (CA1, CB1, CA2, CB2) and Table 3 describes the output handshake modes for CA2 and CB2.

Register Names:
Port A Data Registers (PDA, P1DA, P2DA)

Register Addresses:
$2 (PDA), $0 (P1DA), $1 (P2DA)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

These three registers serve different purposes. PDA is used to read input data and latch data written to the port A output pins P1DA and P2DA are used to read input data and to affect handshake and status activity for PC4/CA1 and PC5/CA2. If enabled, port A input data may be latched into the three port A data registers on an active PC4/CA1 transition as described in **HANDSHAKE OPERATION.**

Description:

Data written into PDA is latched into the port A output latch (see Figure 3) regardless of the state of DDRA Output pins, as defined by DDRA, assume the logic levels of the corresponding bits in the PDA output latch. The PDA output latch allows the user to read the state of the port A output data If the input latch is not enabled, a read of any port A data register reflects the current state of the port A input pins as defined by DDRA and the contents of the output latch for output pins Writes into P1DA or P2DA have no effect upon the output pins or the output data latch. Users are recommended to initialize the port A output latch before changing any pin to an output via the DDRA

MPU accesses of P1DA or P2DA are primarily used to affect handshake and status activity A summary of the effects on the status and warning bits of port A data register accesses is given in Table 4 For more information, see **HANDSHAKE OPERATION** and **Control Register A (CRA)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port A data register

Register Names:
Port B Data Registers (PDB, P1DB, P2DB)

Register Addresses:
$3 (PDB), $C(P1DB), $D (P2DB)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

These three registers serve different purposes The Port B data registers are used to read input data and to latch data written to the port B output pins Writes to PDB and P1DB affect the contents of the output data latch while writes to P2DB do not affect the output data latch P1DB and P2DB accesses additionally affect handshake and status activity for PC6/CB1 and PC7/CB2

Description:

Data written into PDB and P1DB port B registers is latched into the port B output latch (see Figure 3) regardless of the state of DDRB Output pins, as defined by DDRB, assume the logic levels of the corresponding bits in the port B output latch. Reads of any port B data registers reflect the contents of the output data latch for output pins and the current state of the input pins (as determined by DDRB) Users are recommended to initialize the port B output latch before changing any pin to an output via the DDRB

MPU accesses of P1DB or P2DB are primarily used to affect handshake and status activity A summary of the effects on status and warning register bits of port B data register accesses is given in Table 5 For more information, see **HANDSHAKE OPERATION** or **Control Register B (CRB)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port B data register

TABLE 4 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT A DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch Read	Output Latch Write
PDA	None	None	None	Yes	Yes
P1DA	HSA1 cleared to a logic zero	HWA1 loaded into buffer latch	CA2 goes low if output modes 0 or 1 are selected in the CRA	Yes	No
P2DA	HSA2 cleared to a logic zero	HWA2 loaded into buffer latch	CA2 goes low if output modes 0 or 1 are selected in the CRA	Yes	No

TABLE 5 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT B DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDB	None	None	None	Yes	Yes
P1DB	HSB1 cleared to a logic zero	HWB1 loaded into buffer latch	CB2 goes low if output modes 0 or 1 are selected in the CRB	Yes	Yes
P2DB	HSB2 cleared to a logic zero	HWA2 loaded into buffer latch	CB2 goes low if output modes 0 or 1 are selected in CRB	Yes	No

Register Name:

Port C Data Register (PDC)

Register Address:

$4

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

The port C data register (PDC) is used to read input data and to latch data written to the output pins

Description:

Data is written into the port C output latch (see Figure 3) regardless of the state of DDRC Any port C pin defined as a handshake line by the port C function select register (FSR) is not affected by PDC Output pins, as defined by DDRC, assume logic levels of the corresponding bits in the port C output latch. A read of PDC reflects the contents of the output latch for output pins and the current state of the input pins (as reflected in the DDRC) Reset has no effect upon the contents of PDC Users are recommended to initialize the port C output data latch before changing any pin to an output via the DDRC

Register Name:

Data Direction Register for Port A (B) (C)

Register Address:

$6 ($7) ($8)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

Each of the three data direction registers (DDRA, DDRB, and DDRC) define the direction of data flow of the port pins for ports A, B, and C

Description:

A logic zero in a DDR bit places the corresponding port pin in the input mode A logic one in a DDR bit places the corresponding pin in the output mode Any port C pins defined as bidirectional handshake lines also use the port C DDR (DDRC) Input-only handshake lines are not affected by DDRC Reset clears all DDR bits to logic zero configuring all port pins as inputs. The DDRs have no write-inhibit control over the port data output latches Data may be written to the port data registers even though the pins are configured as inputs

Register Name:

Port C Pin Function Select Register (FSR)

Register Address:

$B

Register Bits:

7	6	5	4	3	2	1	0
CFB2	CFB1	CFA2	CFA1	XX	XX	XX	XX

Purpose:

The port C pin function select register defines whether the multifunction port C pins are to operate as "normal" port C lines or as handshake lines

Description:

A logic zero in any FSR bit defines the corresponding port C pin as a "normal" I/O pin. A logic one in any valid FSR bit defines the corresponding port C pin as a handshake line. Pins defined as handshake lines function according to the contents of control register A (CRA) or control register B (CRB) The port C data direction register (DDRC) is valid regardless of FSR contents for all pins except PC4/CA1 and PC6/CB1 Transitions on port C pins not defined as handshake pins do not effect the handshake/interrupt status register Reset clears all FSR bits to a logic zero Users are recommended to initialize the data direction and control registers before modifying the FSR.

Register Name:
Handshake/Interrupt Status Register (HSR)

Register Address:
$E

Register Bits:

7	6	5	4	3	2	1	0
IRQF	XX	XX	XX	HSB2	HSA2	HSB1	HSA1

Purpose:

The handshake interrupt status register is a read-only flag register that may be used during a polling routine to determine if any enabled input handshake transition, as defined by the control register (CRA and CRB), has occurred.

Description:

If an enabled input handshake transition occurs then the appropriate HSR bit (HSB2, HSA2, HSB1, or HSA1) is set. The IRQ flag bit (bit 7, IRQF) is set when one or more of the HSR bits 0-3 and their corresponding control register bits are set to a logic one as shown in the following equation:

$$\text{Bit } 7 = \text{IRQF} = [\text{HSB2}\bullet\text{CRB2(3)}] + [\text{HSA2}\bullet\text{CRA2(3)}] + [\text{HSB1}\bullet\text{CRB1(0)}] + [\text{HSA1}\bullet\text{CRA1(0)}]$$

The numbers in ( ) indicate which bit in the control register enables the interrupt

Handshake/interrupt status register bits are cleared by accessing the appropriate port data register. The following table lists the HSR bit and the port data register that must be accessed to clear the bit

To Clear HSR Bit	Access Register
HSB2	P2DB
HSA2	P2DA
HSB1	P1DB
HSA1	P1DA

Reset clears all handshake/interrupt status register bits to a logic zero.

Register Name:
Handshake Warning Register (HWR)

Register Address:
$F

Register Bits:

7	6	5	4	3	2	1	0
XX	XX	XX	XX	HWB2	HWA2	HWB1	HWA1

Purpose:

The warning register is a read-only flag register that may be used to determine if a second attempt to set a handshake/interrupt status register bit has been made before the original had been serviced

Description:

Each bit in the handshake/interrupt status register, except IRQF, has a corresponding bit in the handshake warning register. If an attempt is made to set a bit in the handshake/interrupt status register that is already set, then the corresponding bit in the handshake warning register is also set An attempt is the occurrence of any enabled input handshake transition as defined by the control registers

A handshake warning register bit is cleared by first reading the appropriate data register then reading the handshake warning register Reading the data register (either P1DA, P2DA, P1DB, or P2DB) loads a buffer latch with the proper bit in the handshake warning register (HWA1, HWA2, HWB1, and HWB2, respectively). The next read of the handshake warning register clears the appropriate bit without affecting the other three handshake warning register bits The upper four bits, HWR4-HWR7, always read as logic zeros. If a port data register is not read before reading the handshake warning register, then the handshake warning register bits will remain unaffected Reset clears all HWR bits to a logic zero.

Recommended status register handling sequence·

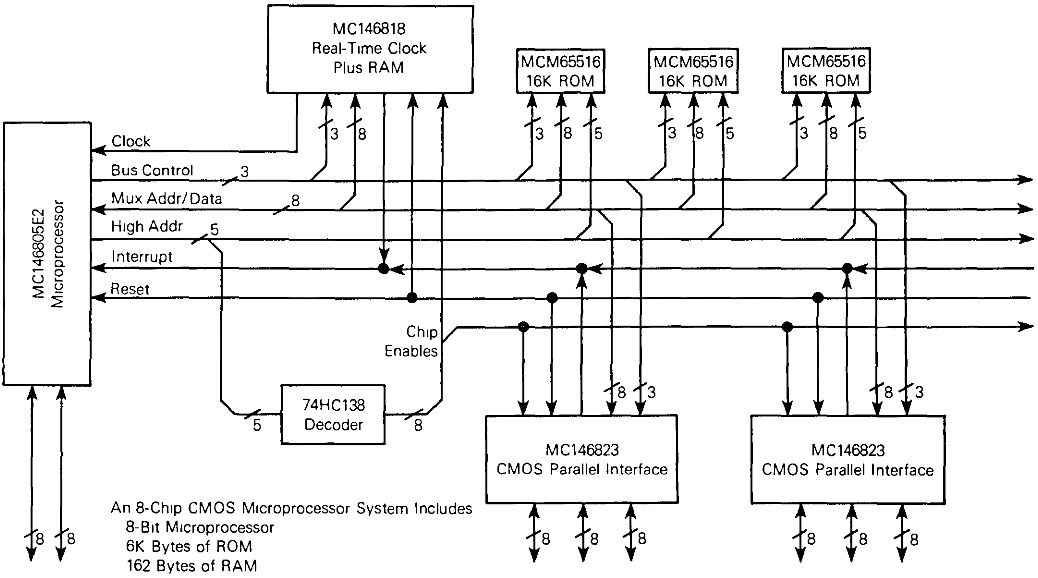
1.	Read status register	(User determines which if any enabled handshake transition occurred)
2	Read/write port data indicated by status register	(Clears associated status bit and latches appropriate warning register bit in the buffer latch)
3	Read warning register	(Latched warning bit is cleared and the remaining bits are unaffected)

TYPICAL INTERFACING

The MC146823 is best suited for use with microprocessors which generate an address-then-data-multiplexed bus. Figure 4 shows the MC146823 in a typical CMOS system that uses the MC146805E2 CMOS MPU. Other multiplexed microprocessors can be used as easily.

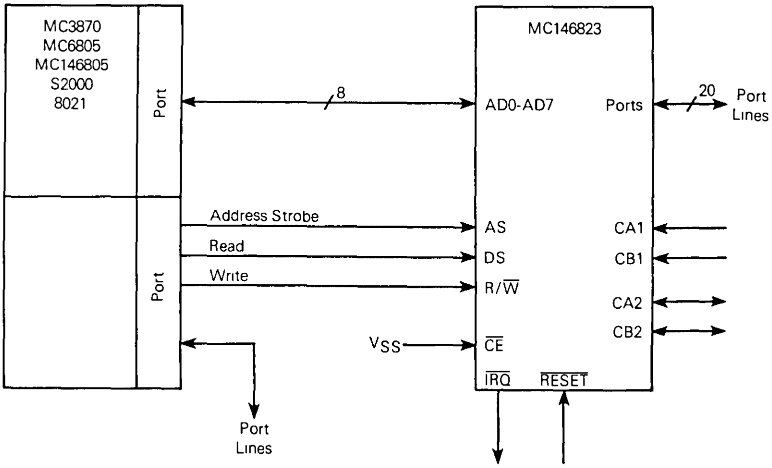
A single-chip microcomputer (MCU) may be interfaced with 11 port lines as shown in Figure 5. This interface also requires some software overhead to gain up to 13 additional I/O lines and the MC146823 handshake lines

FIGURE 4 — A TYPICAL CMOS MICROPROCESSOR SYSTEM


MC146818
Real-Time Clock
Plus RAM
MCM65516
16K ROM
MCM65516
16K ROM
MCM65516
16K ROM
MC146805E2
Microprocessor
Clock
Bus Control
Mux Addr/Data
High Addr
Interrupt
Reset
Chip
Enables
74HC138
Decoder
MC146823
CMOS Parallel Interface
MC146823
CMOS Parallel Interface
3
8
5

An 8-Chip CMOS Microprocessor System Includes
8-Bit Microprocessor
6K Bytes of ROM
162 Bytes of RAM
64 Parallel I/O Pins
RTC Function

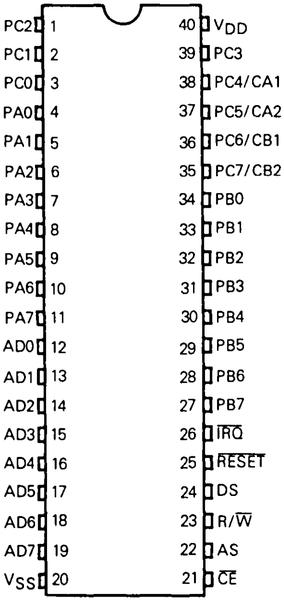
FIGURE 5 — MC146823 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE-CHIP MICROCOMPUTER


MC3870
MC6805
MC146805
S2000
8021
Port
Port
8
Address Strobe
Read
Write
Port
Lines
VSS
MC146823
AD0-AD7
Ports
20
Port
Lines
AS
DS
R/W̄
C̄Ē
ĪR̄Q̄
R̄ĒS̄ĒT̄
CA1
CB1
CA2
CB2

ORDERING INFORMATION ($T_A = 0°C$ to $+70°C$)

Package Type	Order Number
Plastic – P Suffix	MC146823P
PLCC – FN Suffix	MC146823FN

PIN ASSIGNMENT


PC2 1
PC1 2
PC0 3
PA0 4
PA1 5
PA2 6
PA3 7
PA4 8
PA5 9
PA6 10
PA7 11
AD0 12
AD1 13
AD2 14
AD3 15
AD4 16
AD5 17
AD6 18
AD7 19
VSS 20
40 VDD
39 PC3
38 PC4/CA1
37 PC5/CA2
36 PC6/CB1
35 PC7/CB2
34 PB0
33 PB1
32 PB2
31 PB3
30 PB4
29 PB5
28 PB6
27 PB7
26 IRQ
25 RESET
24 DS
23 R/W
22 AS
21 CE

Pin assignments are the same for both the dual-in-line and chip carrier package

MC68HC24

Advanced Information

Port Replacement Unit (PRU)

The MC68HC24 is a peripheral device which replaces ports B and C of the MC68HC11 microcomputer (MCU). These ports are lost when the MCU is placed in the expanded or special test modes of operation. Port B is a general-purpose output port. Port C is a general-purpose input/output port complemented by full handshake capability. This device can also be used in an emulator as a replacement for port B, port C, STRA, and STRB. Applications requiring external memory in early production or top of the line models can also use the MC68HC24 for parallel I/O. When used in these expanded systems, a later switch to a single-chip solution will be transparent to software.

The MC68HC24 is not restricted to simply replacing MC68HC11 ports. The MC68HC24 should be considered as a cost-effective solution for any CMOS microcomputer system requiring I/O expansion, parallel printer interface, or interprocessor communications in multiple MCU systems.

Hardware Features

- Supports All Handshake and I/O Modes of the MC68HC11 Ports
- Automatic Conformance to the MC68HC11 Variable Memory Map
- Multiplexed Address/Data Bus
- Can Be Used with the MC68HC11, MC146805E2, MC146805E3, and other CMOS Microcomputers
- 0- to 2.1-MHz Operation

Software Features

- Software Compatible to MC68HC11 in Single-Chip Mode
- Minimizes Software Overhead for Parallel I/O Handshake Protocols

This document contains information on a new product. Specifications and information herein are subject to change without notice.

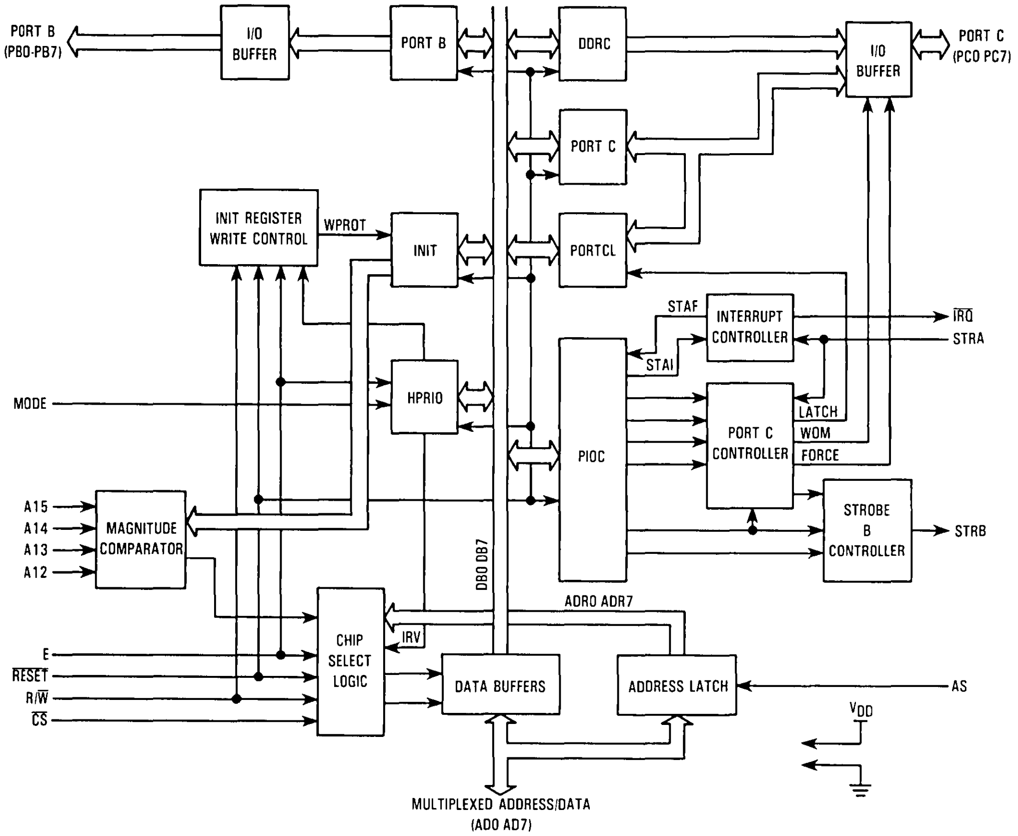

PORT B
(PB0-PB7)
I/O
BUFFER
PORT B
DDRC
I/O
BUFFER
PORT C
(PC0 PC7)
PORT C
INIT REGISTER
WRITE CONTROL
WPROT
INIT
PORTCL
STAF
INTERRUPT
CONTROLLER
IRQ
STRA
STAI
MODE
HPRIO
PIOC
LATCH
PORT C
CONTROLLER
WOM
FORCE
A15
A14
A13
A12
MAGNITUDE
COMPARATOR
STROBE
B
CONTROLLER
STRB
DB0 DB7
ADR0 ADR7
E
RESET
R/W
CS
CHIP
SELECT
LOGIC
IRV
DATA BUFFERS
ADDRESS LATCH
AS
VDD
MULTIPLEXED ADDRESS/DATA
(AD0 AD7)

Figure 1. Block Diagram

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.5 to V_{DD} +0.5	V
Current Drain per Pin	I_{IK}	25	mA
Operating Temperature Range MC68HC24 MC68HC24V MC68HC24M	T_A	T_L to T_H −40 to +85 −40 to +105 −40 to +125	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry which protects the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$ Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 40-Pin DIP Plastic 44-Pin Quad Pack	θ_{JA}	 60 70	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

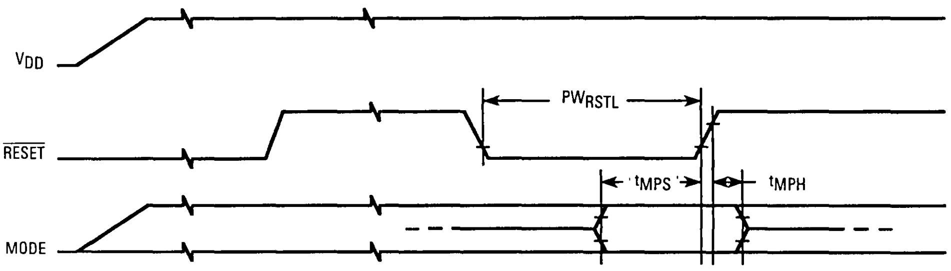
Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

MODE SELECTION AND RESET TIMING (V_{DD} = 5.0 V ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 2)

Characteristic	Symbol	Min	Typ	Max	Unit
$\overline{\text{RESET}}$ Low Input Pulse Width	PW_{RSTL}	2	—	—	E_{cyc}
Mode Programming Setup time	t_{MPS}	2	—	—	E_{cyc}
Mode Programming Hold Time	t_{MPH}	0	—	—	E_{cyc}



NOTE: Measurement points shown are 20% and 70% V_{DD}.

Figure 2. Mode Selection and Reset Timing Diagram

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 V ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} = \pm 10$ μA) All Outputs All Outputs Except $\overline{IRQ}$ (see Note 1)	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	—	0.4	V
Output High Voltage ($I_{Load} = -0.8$ mA, V_{DD} = 4.5 V) All Outputs Except $\overline{IRQ}$ (see Note 1)	V_{OH}	$V_{DD}-0.8$	—	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	$0.2 \times V_{DD}$	V
Input High Voltage All Inputs	V_{IH}	$0.7 \times V_{DD}$	V_{DD}	V
I/O Ports, 3-State Leakage ($V_{in} = V_{IH}$ or V_{IL}) PB0-PB7, PC0-PC7, AD0-AD7	I_{OZ}	—	±10	μA
Input Current ($V_{in} = V_{DD}$ or V_{SS}) E, AS, R/$\overline{W}$, $\overline{CS}$, MODE, A12–A15, IOTEST, STRA	I_{in}	—	±1	μA
Total Supply Current (see Note 2)	I_{DD}	—	5	mA
Input Capacitance E, AS, R/$\overline{W}$, $\overline{CS}$, MODE, A12–A15, IOTEST, STRA PB0–PB7, PC0–PC7, AD0–AD7	C_{in}	— —	8.0 12.0	pF
Power Dissipation	P_D	—	28	mW

NOTES:

1. V_{OH} specification for $\overline{IRQ}$ is not applicable because it is an open-drain output pin.
2. Test conditions for total supply current are as follows:
   a. C_L = 90 pF on Port B and AD0 through AD7, no dc loads, t_{cyc} = 500 ns.
   b. Port C programmed as inputs.
   c. $V_{IL} = V_{SS} + 0.2$ V for $\overline{PC0\text{-}PC7}$, $\overline{AD}$7-AD2 and AD0 (during E = V_{IL}), $\overline{CS}$
   $V_{IH} = V_{DD} - 0.2$ V for $\overline{RESET}$, R/$\overline{W}$, AD1 (during E = V_{IL}), MODE.
   d. The E input is a squarewave from $V_{SS}+0.2$ V to $V_{DD}-0.2$ V.
   e. AS input is 25% duty cycle from $V_{SS}+0.2$ V to $V_{DD}-0.2$ V.

PERIPHERAL PORT TIMING (V_{DD} = 5.0 V ± 10%, all timing is shown with respect to 20% V_{DD} and 70% V_{DD})

Characteristic	Symbol	Min	Max	Unit	Figure No.
Peripheral Data Setup Time (Port C)	t_{PDSU}	100	—	ns	4
Peripheral Data Hold Time (Port C)	t_{PDH}	50	—	ns	4
Delay Time, E Negative Transition to Peripheral Data Valid (Ports B and C, see Note 1)	t_{PWD}	—	100	ns	3, 5, 8, 9
Input Data Setup Time (Port C)	t_{IS}	60	—	ns	6, 7
Input Data Hold Time (Port C)	t_{IH}	100	—	ns	6, 7
Delay Time, E Positive Transition to STRB Asserted (see Note 1)	t_{DEB}	—	80	ns	5, 7, 8, 9
Setup Time, STRA Asserted to E Negative Transition (see Note 2)	t_{AES}	0	—	ns	7, 8, 9
Delay Time, E Rise to $\overline{IRQ}$ (see Note 3)	t_{IRQD}	—	60	ns	7, 8, 9
Delay Time, STRA Asserted to Port C Data-Out Valid (see Note 4)	t_{PCD}	—	100	ns	9
Hold Time, STRA Negated to Port C Data	t_{PCH}	10	—	ns	9
Three-State Hold Time	t_{PCZ}	—	150	ns	9
STRA Cycle Time	t_{Scyc}	2	—	E_{cyc}	6, 7

NOTES:

1. The referenced clock edge for this characteristic differs from the MC68HC11.
2. If this setup time is met, STRB will be acknowledged in the next cycle. If it is not met, the response will be delayed one more cycle.
3. $\overline{IRQ}$ active when STAI is set in PIOC.
4. Port C timing is only valid for active drive (CWOM bit is not set in PIOC).

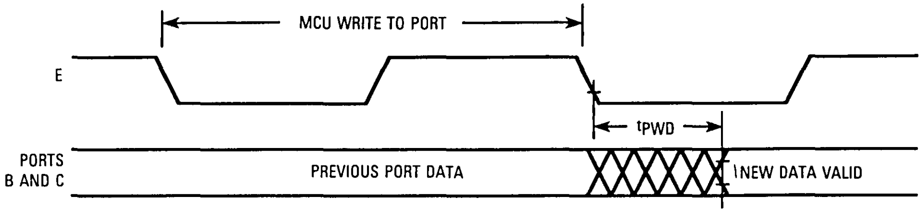
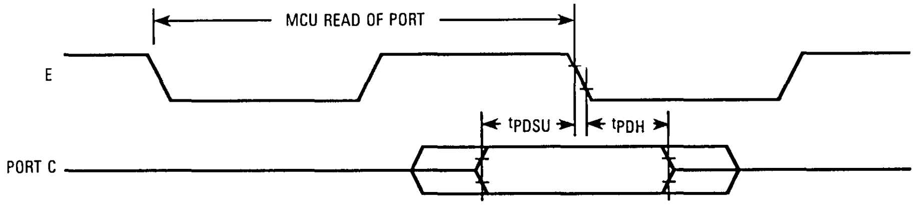

MCU WRITE TO PORT
E
tPWD
PORTS B AND C
PREVIOUS PORT DATA
NEW DATA VALID

Figure 3. Port Write Timing Diagram


MCU READ OF PORT
E
tPDSU
tPDH
PORT C

NOTE: For Non-Latched Operation of Port C

Figure 4. Port C Static Read Timing Diagram

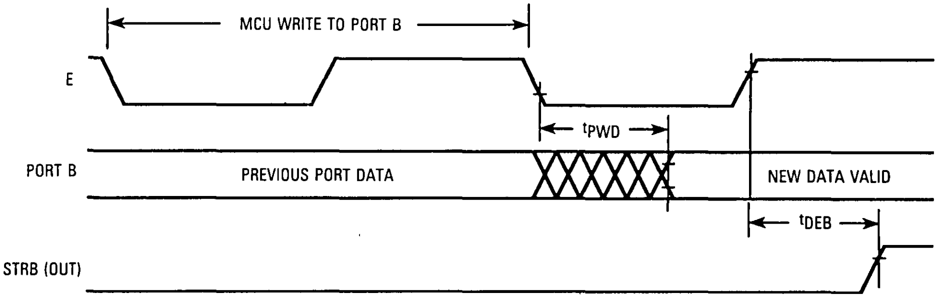

MCU WRITE TO PORT B
E
tPWD
PORT B
PREVIOUS PORT DATA
NEW DATA VALID
tDEB
STRB (OUT)

Figure 5. Simple Output Strobe Timing Diagram

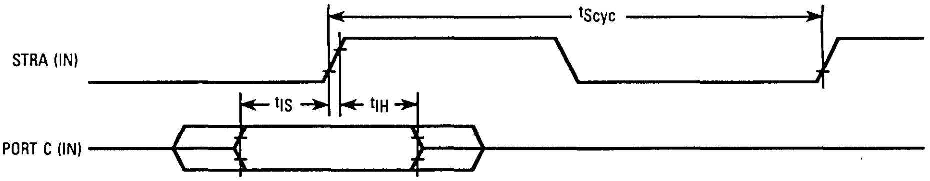
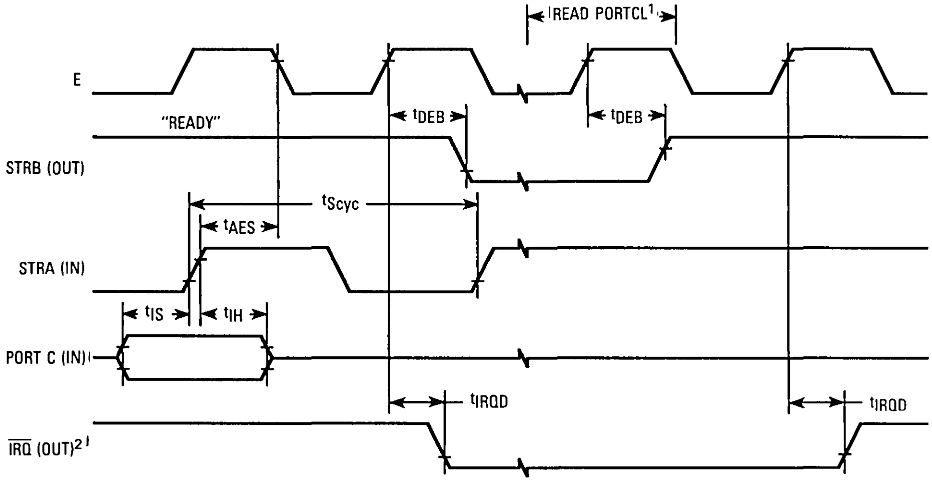
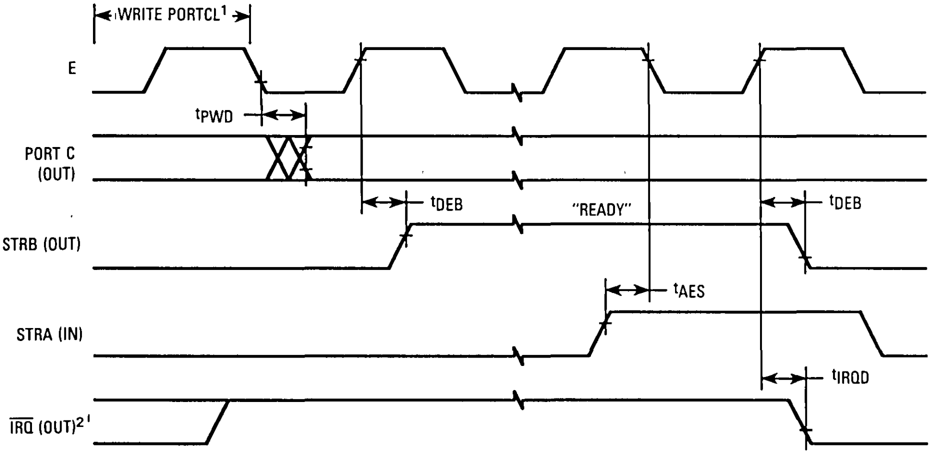

tScyc
STRA (IN)
tIS
tIH
PORT C (IN)

Figure 6. Simple Input Strobe Timing Diagram


READ PORTCL¹
E
"READY"
tDEB
tDEB
STRB (OUT)
tScyc
tAES
STRA (IN)
tIS
tIH
PORT C (IN)
tIRQD
tIRQD
IRQ (OUT)²

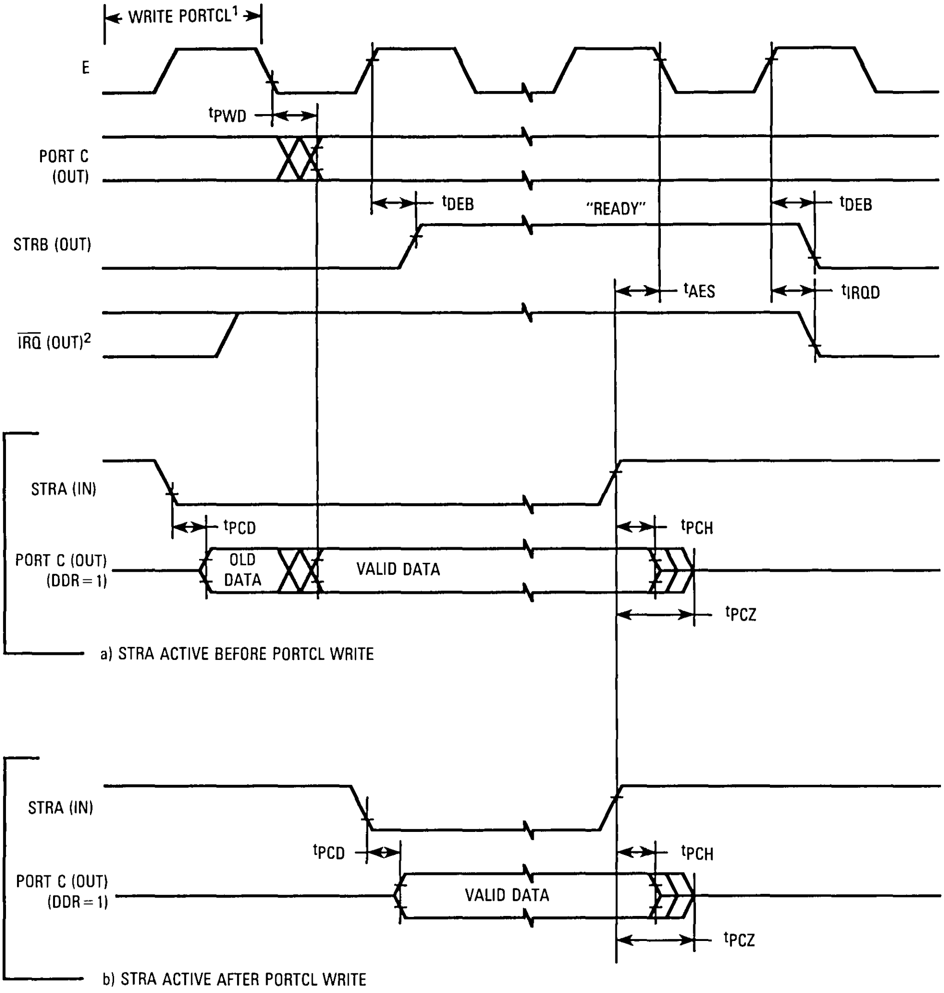
NOTES:
1. After reading PIOC with STAF set.
2. STAI set in PIOC.
3. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 7. Port C Input Handshake Timing Diagram


WRITE PORTCL¹
E
tPWD
PORT C (OUT)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
STRA (IN)
tIRQD
IRQ (OUT)²

NOTES:
1. After reading PIOC with STAF set.
2. STAI set in PIOC.
3. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 8. Port C Output Handshake Timing Diagram


WRITE PORTCL1
E
tPWD
PORT C (OUT)
tDEB
"READY"
tDEB
STRB (OUT)
tAES
tIRQD
IRQ (OUT)2
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
OLD DATA
VALID DATA
tPCZ
a) STRA ACTIVE BEFORE PORTCL WRITE
STRA (IN)
tPCD
tPCH
PORT C (OUT) (DDR=1)
VALID DATA
tPCZ
b) STRA ACTIVE AFTER PORTCL WRITE

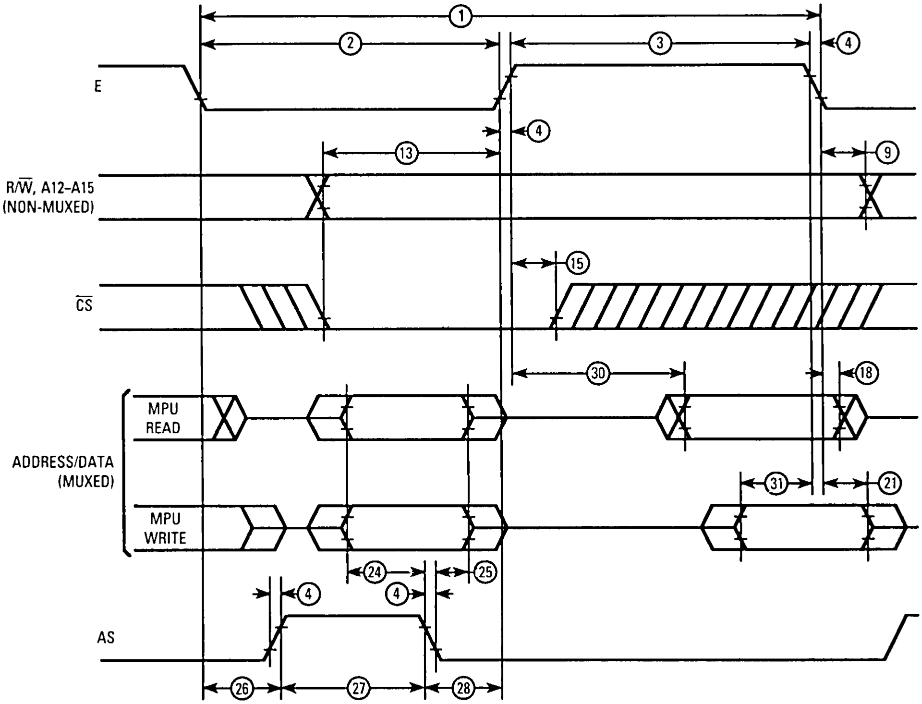
NOTES:

1. After reading PIOC with STAF set.
2. STAI set in PIOC.
3. Figure shows rising edge STRA (EGA=1) and high true STRB (INVB=1).

Figure 9. Three-State Variation of Output Handshake Timing Diagram
(STRA Enables Output Buffer)

BUS TIMING CHARACTERISTICS (V_{DD} = 5.0 V ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; see Figure 10 for detailed timing diagrams)

Ident. Number	Characteristic	Symbol	1 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1000	—	476	—	ns
2	Pulse Width, E Low	PW_{EL}	460	—	215	—	ns
3	Pulse Width, E High	PW_{EH}	450	—	210	—	ns
4	Input and Clock Rise and Fall Time	t_r, t_f	—	25	—	20	ns
9	Address Hold Time	t_{AH}	20	—	10	—	ns
13	Non-Muxed Address Setup Time before E	t_{AS}	100	—	50	—	ns
15	Chip Select Hold Time ($\overline{CS}$)	t_{CSH}	20	—	20	—	ns
18	Read Data Hold Time	t_{DHR}	10	75	10	75	ns
21	Write Data Hold Time	t_{DHW}	10	—	10	—	ns
24	Muxed Address Valid Time to AS Fall	t_{ASL}	60	—	20	—	ns
25	Muxed Address Hold Time	t_{AHL}	40	—	20	—	ns
26	Delay Time, E Fall to AS Rise	t_{ASD}	60	—	30	—	ns
27	Pulse Width, AS High	PW_{ASH}	150	—	75	—	ns
28	AS Fall to E Rise	t_{ASED}	60	—	30	—	ns
30	Peripheral Output Data Delay Time from E Rise (Read)	t_{DDR}	20	240	10	120	ns
31	Peripheral Data Setup Time (Write)	t_{DSW}	150	—	60	—	ns



NOTE: Measurement points shown are 20% and 70% V_{DD}.

Figure 10. Bus Timing Diagram

PIN DESCRIPTION

The input and output pins for the port replacement unit are described in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the peripheral using these two pins. Power is V_{DD} and ground is V_{SS}.

RESET ($\overline{RESET}$)

This active-low control input pin is used to initialize the MC68HC24 to a known start-up state. The system state after a reset is detailed in **STATE AFTER RESET**. This pin must remain at a low level for a minimum of two E-clock cycles to be recognized.

ENABLE (E)

The E clock input is the basic MPU/MCU clock. This clock provides most timing reference information to the MC68HC24. In general, when E is low, an internal process is taking place. When E is high, data is being accessed.

The E-clock runs at the external bus rate of the MPU/MCU and may range in frequency from dc to the maximum operating frequency of the device (i.e., this peripheral part is static). More information on the timing relationships between the various signals may be found in **PERIPHERAL PORT TIMING** and **BUS TIMING CHARACTERISTICS.**

ADDRESS STROBE (AS)

The AS input pulse serves to demultiplex the address/data bus. The falling edge of AS causes the addresses AD0 through AD7 to be latched within the MC68HC24.

READ/WRITE (R/$\overline{W}$)

The read/write pin is a high-impedance input signal which is used to control the direction of data flow along the multiplexed address/data bus. When the device is selected and the R/$\overline{W}$ input is high, the data output buffers are enabled and a selected register is read.

Data is written into the selected register when the chip is selected with R/$\overline{W}$ low. R/$\overline{W}$ signal is not latched by the MC68HC24. In order to guarantee that register contents are not corrupted, R/$\overline{W}$ must be stable prior to the rising edge of the E clock and must remain stable throughout the E clock high time.

CHIP SELECT ($\overline{CS}$)

This input pin serves as the device chip select. The MC68HC24 is selected when 1) $\overline{CS}$ is low, 2) the contents of the INIT register match address lines A12 through A15, and 3) the lower order address lines (AD0 through AD7) select an internal register address. All three of these conditions must be met to access the internal registers. The $\overline{CS}$ signal is latched on the rising edge of the E clock and must be stable prior to that edge.

No action will take place within the MC68HC24 during bus cycles in which 1) $\overline{CS}$ is not asserted, 2) the A12 through A15 address lines do not match the contents of the INIT register, or 3) an internal register is not addressed.

ADDRESS AND DATA BUS (AD0 through AD7)

Multiplexed bus microprocessors save pins by presenting the address during the first portion of the bus cycle and using those same pins during the second portion of the bus cycle for data. Address and data multiplexing does not slow the access time of the MC68HC24 since the bus reversal from address to data occurs during the internal register access time.

The low-order address must be stable (valid) prior to the fall of AS at which time the MC68HC24 latches the address present on AD0 through AD7. If the latched address is decoded, if $\overline{CS}$ is asserted, and if A12 through A15 match the contents of the INIT register, a selected register will be accessed.

Although a 64-byte register block is reserved for the registers, only seven of the locations are currently implemented. See **INTERNAL REGISTER DESCRIPTION** for details about specific addresses.

Valid write data must be presented by the MPU/MCU during the E high period of the write cycle. In a read cycle, the MC68HC24 outputs eight bits of data during the second half of the read bus cycle and then ceases driving the bus (returns to a high-impedance state) after the falling edge of E.

HIGH-ORDER ADDRESS (A12 through A15)

The address lines, A12 through A15, are the nonmultiplexed high-order address lines of the MPU/MCU. These signals are used internally to establish a partial decoding for the chip select. They are latched by the rising edge of the E clock and must be stable prior to this edge. A magnitude comparator checks the value of these lines against a value stored in the INIT register. If they match, $\overline{CS}$ is asserted, and an internal register is addressed, the device will be accessed during the current bus cycle.

PORT B (PB0 through PB7)

Port B (PB0 through PB7) is an 8-bit general purpose output port. In the simple strobed mode of operation, STRB is pulsed for each write to port B. See **I/O PORTS** for more information.

PORT C (PC0 through PC7)

Each line of port C is individually programmable as either an input or an output via its data direction register (DDRC). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. Several handshake modes are available on this port (see **I/O PORTS**).

STROBE A (STRA)

Strobe A is an edge detecting input used by port C. In the simple strobed and input handshake modes of operation, the programmed edge on STRA will latch the data on the port C inputs into PORTCL. In the output handshake mode, STRA is an edge-sensitive acknowledge input signal indicating that port C output data has been accepted by the external device.

STROBE B (STRB)

While operating in the simple strobed I/O mode, Strobe B is a strobe output which pulses for each write to port B. In the full handshake mode of parallel I/O, STRB acts as a handshake output line. The STRB pin is a READY output in the input handshake mode, inhibiting the external device from strobing new data into port C. In the output handshake mode, STRB is again a READY output; however, in this case it indicates that new data has been written to port C by the microprocessor.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ output pin is an open-drain active-low signal that may be used to interrupt the microprocessor with a service request. The open drain output allows multiple devices to be wire-ORed together. This configuration requires an external resistor to V_{DD} as no internal pullup is provided.

The MC68HC11 I/O port interrupts share the same vector address as $\overline{\text{IRQ}}$. As a result, an expanded MC68HC11 system incorporating an MC68HC24 (to replace the displaced I/O features) will appear to the software as a single chip solution. Refer to the **INTERNAL REGISTER DESCRIPTION—PIOC** and **I/O PORTS—FULL HANDSHAKE I/O** for additional information.

I/O TEST (IOTEST)

This is a factory test feature and the IOTEST pin must be tied directly to V_{SS} for normal operation.

I/O PORTS

There are two 8-bit parallel I/O ports on the MC68HC24. Port B is a general purpose output-only port, whereas port C may be used as general purpose input and/or output pins as specified by DDRC. In conjunction with STRA and STRB, ports B and C may be used for special strobed and handshake modes of parallel I/O as well as general purpose I/O.

GENERAL PURPOSE I/O (PORT C)

When used as general purpose I/O signals, each bit has associated with it one bit in the PORTC data register and one bit in the corresponding position in the data direction register (DDRC). The DDRC is used to specify the primary direction of data on the I/O pin; however, specification of a line as an output does not disable the ability to read the line as a latched input.

When a bit which is configured as an output is read, the value returned will be the value at the input to the pin driver. When a pin is configured as an input (by clearing the DDRC bit) the pin becomes a high-impedance input. When writing to a bit that is configured as an input, the value will not affect the I/O pin; however, the bit will be stored to an internal latch so that if the line is later recognized as an output this value will appear at the I/O pin.

This operation can be used to preset a value for an output port prior to configuring it as an output, so that glitches of an output state which are not defined for the external system may be avoided. Reset configures the port for input by clearing both the DDR and the data register.

FIXED DIRECTION I/O (PORT B)

Port B is a general purpose output-only port. The data direction is fixed in order to properly emulate the operation of the MC68HC11 port B. Reads of port B return the levels sensed at the input of the pin drivers. Write data is stored in an internal latch which directly drives the output pin driver. Reset clears the data register forcing the outputs low.

SIMPLE STROBED I/O

The simple strobed mode of parallel I/O is controlled by the parallel I/O control (PIOC) register. This mode is selected when the HNDS bit in the PIOC register is clear. This mode forces PORTCL to be a strobed input port with the STRA pin used as the edge detecting latch command input. Also, port B becomes a strobed output port with the STRB pin acting as the output strobe.

Strobed Input Port C

In this mode, there are two addresses where port C may be read—PORTC data register and PORTCL latch register. Even when the strobed input mode is selected, one or all of the bits in port C may be used as general purpose I/O lines. In other words, the DDRC register still controls the data direction of all port C pins.

The STRA pin is used as an edge-detecting input. Either falling or rising edges may be specified as the significant edge by use of the EGA bit in PIOC. Whenever the selected active edge is detected at the STRA pin, the current logic levels at port C are latched into the PORTCL register and the strobe A flag (STAF) bit in PIOC is set.

If the STAI bit in PIOC is also set, then an interrupt sequence is requested on the $\overline{\text{IRQ}}$ pin. The STAF flag is automatically cleared by reading the PIOC register (with STAF set) followed by a read of the PORTCL register. Additional active edges of STRA continue to latch new data into PORTCL regardless of the state of the STAF flag. Consecutive active edges on STRA must be a minimum of two E-clock cycles apart.

Reads of the PORTCL register return the last value latched, while reads of PORTC return the static level of the port C pins (inputs) or the level at the input to the pin driver (outputs).

Strobed Output Port B

In this mode, the STRB pin is a strobe output which is pulsed each time there is a write to port B. Data written to PORTB is stored in a latch which drives the port B pin drivers. Reads of port B return the levels at the inputs of those pin drivers.

The INVB bit in the PIOC register controls the polarity of the pulse out of the STRB pin. If the INVB bit is set, the strobe pulse will be a high going pulse (two E-clock periods long) on a normally low line. If the INVB bit is clear, the strobe pulse will be low-going pulse (two E-clock periods long) on a normally high line.

FULL HANDSHAKE I/O

The full handshake modes of parallel I/O involve port C, STRA, and STRB. There are two basic modes (input and output) and an additional variation on the output handshake mode that allows for three-state operation of port C. In all handshake modes, STRA is an edge detecting input and STRB is a handshake output line. The effect of DDRC is discussed in **Input Handshake Protocol, Output Handshake Protocol, Three State Variation,** and **Interaction of Handshake and General Purpose I/O.**

Input Handshake Protocol

In the handshake scheme, port C is a latching input port, STRA is an edge-sensitive latch command from the external system that is driving port C and STRB is a READY output line controlled by logic in the MC68HC24.

In a typical system, an external device wishing to pass data to port C would test the READY line (STRB). When a ready condition was recognized, the external device would place data on the port C inputs followed by a pulse on the STRA input to the MC68HC24. The active edge on the STRA line would latch the port C data into the PORTCL register, set the STAF flag (optionally causing an interrupt), and deassert the READY line (STRB). Deassertion of the READY line would automatically inhibit the external device from strobing new data into port C. Reading the PORTCL latch register, after reading PIOC with STAF set, clears the STAF flag. Whenever PORTCL is read, the READY (STRB) line is asserted indicating that new data may now be strobed into port C.

The STRB line can be configured (with the PLS control bit) to be a pulse output (pulse mode) or a static output (interlocked mode). The only difference between the pulse and interlock modes is that in pulse mode, the READY line pulses (asserts) for only two E-clock periods after the latched data becomes available. While in interlock mode, the asserted state of the READY line lasts until new data is strobed into port C via the STRA input line.

The port C DDR bits should be cleared (input) for each bit that is to be used as a latched input bit. It is, however, possible to use some port C bits as latched inputs with the input handshake protocol and at the same time use other port C bits as static inputs and still other port C bits as static output bits.

The input handshake protocol has no effect on the use of port C bits as static inputs or static outputs. Reads of the PORTC register always return the static logic level at the port C pins (for lines configured as input) or at the inputs to the pin drivers (for lines configured as outputs). Data latched into PORTCL always reflects the level at the port C pins. Writes to either the PORTC address or the PORTCL address will write information to the port C output register without affecting the input handshake strobes.

NOTE

> After programming PIOC to enter the input handshake mode, STRB will remain in the inactive state. This precaution has been taken to ensure that the external system will not strobe data into PORTCL before all intialization is complete. When ready to accept data, the MPU/MCU should perform a dummy read of the PORTCL address. This operation will assert STRB initiating the input handshake protocol.

Output Handshake Protocol

In the output handshake scheme, port C is an output port, STRB is a READY output, and STRA is an edge-sensitive acknowledge input signal indicating that port C output data has been accepted by the external device. In a variation of this output handshake operation, STRA is used as an output enable input as well as an edge-sensitive acknowledge input.

In a typical system, the controlling processor writes to the MC68HC24, placing data in the port C output latch. Stable data on the port C pins is indicated by the automatic assertion of the MC68HC24 READY (STRB) line. The external device then processes the available data and pulses the STRA input to indicate that new data may be placed on the port C output lines. The active edge on STRA causes the READY (STRB) line to be automatically deasserted and the STAF status flag to be set (optionally causing an interrupt). In response to STAF being set, the program puts out new data on port C as required.

There are two addresses associated with the port C data register, the normal PORTC data address and a second address (PORTCL) that accesses the input latch on reads and the normal port on writes. On writes to the second address (PORTCL), the data goes to the same port output register as it would on a write to the PORTC address but the STAF flag bit is cleared (provided PIOC was first read with the STAF bit set). This allows an automatic clearing mechanism in output handshake modes to co-exist with normal port C outputs.

All eight bits in port C must be used as outputs while the output handshake protocol is selected. That is, part of port C may not be used for static or latched inputs while the remaining bits are being used for output handshake. The following paragraphs cover this limitation in more detail.

Output Handshake Protocol, Three-State Variation

There is a variation to the output handshake protocol that allows three-state operation of port C. It is possible to directly interconnect this 8-bit parallel port to other 8-bit three-state devices with no additional external parts.

The STRA signal is used as an acknowledge/enable input whose sense is controlled by the EGA bit in the PIOC register. The EGA bit specifies the transition from the asserted to the deasserted state of the STRA input signal. If EGA is zero, the asserted state is high and falling edges are interpreted as acknowledge signals. If EGA is one, the asserted state is low and rising edges are interpreted as acknowledge signals.

As long as the STRA input pin is negated, all port C bits obey the data direction specified by DDRC. Bits which are configured as inputs (DDR bit equals zero) will be high impedance. When the STRA input is asserted, all port C lines are forced to be outputs regardless of the data in DDRC.

This operation limits the ability to use some port C bits as static inputs while using others as handshake outputs. However, it does not interfere with the use of some port C bits as static outputs while others are being used as three-state handshake outputs. Port C bits which are to be used as static outputs or normal handshake outputs should have their corresponding DDRC bits set. Bits which are to be used as three-state handshake outputs should have their corresponding DDRC bits clear.

Interaction of Handshake and General Purpose I/O

There are two addresses associated with the port C data register: the normal PORTC address and a second address (PORTCL) that accesses the input latch on reads and the normal port on writes. On writes to the second address (PORTCL), the data goes to the same port output register as it would on a write to the port output address. When operating in the output handshake mode, writing to PORTC will not clear the STAF bit whereas writing to PORTCL will clear it. This allows an automatic clearing mechanism to co-exist with normal port C outputs.

When full input handshake protocol is specified, both general purpose input and/or general purpose output can co-exist at port C. However, the three-state feature of the output handshake mode interferes with general purpose inputs in two ways.

First, in full output handshake, the port C pins are forced to be driven outputs during any period in which STRA is in its active state regardless of the state of the DDRC bits. This potentially conflicts with any device trying to drive port C unless the external device has an open-drain type output driver.

Secondly, the value returned on reads of port C is the state at the inputs to the pin drivers regardless of the state of the DDRC bits. This allows data written for output handshake to be read even if the pins are in a three-state condition.

The following is an example of port C being used for full input handshake, general purpose input, and general purpose output all at the same time. Assume that the PIOC and DDRC control registers are set up as follows:

```
PIOC-0111 0000    /STAF/STAI/CWOM/HNDS/ /OIN/PLS/EGA/INVB/
DDRC-0000 1100    /MSB...    ...LSB/
```

In this example, port C bits b7 through b4 will be used for input handshake, bits b3 and b2 will be used as open-drain type general purpose outputs, and bits b0 and b1 will be used as general purpose inputs. The DDRC register is configured such that bits b2 and b3 are outputs and the rest of the port C bits are inputs. The PIOC register is configured such that full-input handshake is specified (HNDS equals one and OIN equals zero). CWOM equals one so any pins in port C which are configured as outputs will behave as open-drain type outputs. The other bits in PIOC are not important for the discussion of this example.

When data is latched into PORTCL according to the input handshake protocol, all eight bits are captured although only the four MSBs are of interest to the input handshake software. The data latched into all eight bits of PORTCL will be the levels present at port C pins.

Software driving the bits b2 and b3 general-purpose outputs would perform writes to PORTC which would not affect the handshake protocol or the latching of data into PORTCL. Data written to port C bits b0, b1, and b4 through b7 would also be latched into the internal port C output latch but since the corresponding DDRC bits are zeros, the corresponding port C pins would remain unaffected.

Bit manipulation and read-modify-write instructions could be used on PORTC because reads of PORTC do not affect the input handshake functions. Although writes to PORTCL would also cause data to be written to port C, this address should not be used for general purpose output. This is because bit manipulation and read-modify-write instructions read the location before writing to it and this read would interfere with the input handshake protocol.

Finally, to use bits 0 and 1 for general purpose inputs, simply read PORTC which will return the desired information and will not interfere with the input handshake protocol. Note that the current state of the port C bits b4 through b7 are also read; therefore, even the pins which are being used for input handshake can be read at any time without disturbing the input handshake function.

INTERNAL REGISTER DESCRIPTION

A 64-byte address space is reserved for internal register access, although not all 64 addresses are used. The ABSOLUTE locations where these addresses will appear are specified by the reset initialization software and chip select logic provided by the end user (see INIT register). The following list summarizes the register mnemonics and their associated addresses.

$xx02	PIOC	PARALLEL I/O CONTROL REGISTER
$xx03	PORTC	I/O PORT C
$xx04	PORTB	OUTPUT PORT B
$xx05	PORTCL	ALTERNATE LATCHED PORT C
$xx07	DDRC	PORT C DATA DIRECTION REGISTER
$xx3C	HPRIO	HIGHEST PRIORITY I BIT INTERRUPT AND MISCELLANEOUS
$xx3D	INIT	I/O MAPPING REGISTER

SPECIFIED BY CHIP SELECT DECODING

SPECIFIED BY BITS 0 THROUGH 3 OF THE INIT REGISTER

PARALLEL I/O CONTROL REGISTER (PIOC)

The PIOC register is an 8-bit read/write register except for bit 7 which is a read-only flag bit.

b7	b6	b5	b4	b3	b2	b1	b0	
STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	$xx02
0	0	0	0	0	0	1	1	RESET

b7, STAF — The STAF (strobe A interrupt status flag) bit is set when a selected active edge is detected by the STRA input pin. If b6 (STAI) is set, then an interrupt sequence using the $\overline{IRQ}$ output pin will also be requested whenever the STAF flag is set. This bit is cleared by reset to indicate no interrupt request is pending.

There is an automatic clearing mechanism on this flag bit (STAF) which depends on the operating mode selected. There are three basic strobed modes (see b4, HNDS and b3, OIN).

When HNDS is zero, the simple strobed mode is specified and the OIN bit has no meaning or effect. In this mode, the STAF flag is automatically set by detection of the selected edge on the STRA input pin indicating that new data is available in the port C latch. The

STAF flag is automatically cleared by a read of the PIOC register (with STAF set) followed by a read of the PORTCL latch register.

When HNDS is one and OIN is zero, the input handshake mode is specified. In this mode, the STAF flag is automatically set by detection of the selected edge on the STRA input pin indicating that new data is available in the port C latch. The STAF bit is automatically cleared by a read of the PIOC register (with STAF set) followed by a read of the PORTCL latch register.

When HNDS is one and OIN is one, the output handshake mode is specified. In this mode, the STAF flag is automatically set by detection of the selected edge on the STRA input pin indicating that data from port C has been accepted by the external system. The STAF flag is automatically cleared by a read of the PIOC register (with STAF set) followed by a write to the PORTCL latch register.

b6, STAI The STAI (strobe A interrupt enable mask) bit is used to specify whether or not a hardware interrupt sequence is to be requested whenever STAF is set. To request a hardware interrupt, both the STAI interrupt enable bit and the STAF flag bit must be set. This bit is cleared by $\overline{\text{RESET}}$ so that parallel I/O interrupts are inhibited. The user must write this bit to a one in order to use the strobed and handshake I/O functions in an interrupt-driven, rather than a polled, environment.

b5, CWOM When the CWOM (port C wire-OR mode) bit is zero, the port C output pins operate normally. When this bit is set to one, the port C outputs behave as open-drain type drivers allowing wired-OR type external connections. When CWOM equals one, the top driver device is disabled so that pins may be driven low by writing zeros or become three-state by writing ones. With an external pull-up resistor, the non-driven lines are pulled to logic ones.

This permits port C output pins to be safely wired in parallel with similar CMOS output drivers without fear of contentions which could otherwise cause destructive latch-up. This bit is cleared by $\overline{\text{RESET}}$ so port C pins which are configured as outputs will operate normally.

b4, HNDS When HNDS (handshake mode) bit is clear, the STRA pin acts as a simple input strobe to latch incoming data into the PORTCL latch register and the STRB pin acts as a simple output strobe that pulses after any write to port B. When HNDS is set, it specifies that a handshake protocol involving port C, STRA, and STRB is in effect. In all modes, STRA is an edge-sensitive input and STRB is an output. This bit is cleared by $\overline{\text{RESET}}$. The strobe and handshake modes are described in greater detail in **I/O PORTS**.

b3, OIN The OIN (output or input handshake) bit has no meaning or effect unless HNDS is set to one. When this bit is zero, input handshake protocol is specified. When this bit is a one, output handshake protocol is specified. See **I/O PORTS** for a more detailed description of the handshake protocols.

b2, PLS The PLS (pulse/interlocked handshake) bit has no meaning or effect unless HNDS is set to one. When this bit is zero, interlocked handshake operation is specified. When this bit is one, pulse mode handshake operation is specified.

In interlocked modes, the STRB output line, once activated, remains active indefinitely until the selected edge is detected on the STRA input line. In pulse modes, the STRB output line, once activated, remains active for only two MCU E-clock cycles and then automatically reverts to the inactive state. This bit is cleared by $\overline{\text{RESET}}$. For more details on the handshake protocols, see **I/O PORTS**.

b1, EGA The EGA (active edge for STRA) bit is used to specify which edge (rising or falling) on the STRA input pin is to be considered the active edge. When this bit is zero, the active edge is the falling edge and when this bit is one, the active edge is the rising edge. This bit is set to one by $\overline{\text{RESET}}$.

When output handshake mode is specified, this bit is used to control the PORTC three-state variation as well as select the active acknowledge edge. In the three-state variation, the EGA bit specifies the trailing edge polarity for the STRA input pin which is interpreted as the enable/acknowledge signal. Assertion of STRA overrides the DDRC specification to force port C to be outputs and the edge of negation is the active edge acknowledge command.

If EGA is zero, the falling edge at STRA is the active edge which causes STAF to be set and STRB to be negated. Additionally, if EGA is zero, port C bits obey the DDRC specification while STRA is low but port C is forced to be an output when STRA is high.

If EGA is one, the rising edge at STRA is the active edge. This causes STAF to be set and STRB to be negated. In addition, port C bits obey the DDRC specification while STRA is high, but port C is forced to be an output when STRA is low.

b0, INVB The INVB (Invert Strobe B) bit is used to specify whether or not to invert the normal strobe B (STRB) logic output levels. When this bit is one, no inversion is specified and the active level on the strobe B output line is logic one. When this bit is zero, inversion is specified and the active level on the strobe B output line is logic zero. This bit is set to one by $\overline{\text{RESET}}$ so that the STRB output will initially be in the low state out of reset. For a more detailed description of the handshake protocols, see the **I/O PORTS** section.

PORT C DATA REGISTER (PORTC)

b7	b6	b5	b4	b3	b2	b1	b0	
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	$xx03
0	0	0	0	0	0	0	0	RESET

Port C (PORTC) is a general purpose input/output port complemented by full handshake capability. For bits that are configured as inputs, reads of this address return the level sensed at the pin. For bits configured as outputs, reads return the level sensed at the input to the pin driver. When a port C pin is being used for the three-state variation of parallel output handshake, reads return the level sensed at the input to the pin driver even if the DDR bits suggest that the pin is configured as an input.

Writes to port C cause the value to be latched in the 8-bit port C data register. (Note that this is not the same register as the PORTCL latch register described later.) When the corresponding DDRC bit is set, the value in the port C data register is driven out of the port C pin. This data latch allows the programmer to initialize the data prior to turning on the output drivers by setting bits in the DDRC. The PORTC register is cleared by $\overline{\text{RESET}}$.

PORT B (PORT B DATA REGISTER)

b7	b6	b5	b4	b3	b2	b1	b0	
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	$xx04
0	0	0	0	0	0	0	0	RESET

Port B (PORTB) is a general purpose output-only port. Reads of this address return the level sensed at the input to the pin driver. Writes to Port B cause the value to be latched in the 8-bit Port B data register. The PORTB register is set to zero by $\overline{\text{RESET}}$.

PORT C LATCHED DATA REGISTER (PORTCL)

b7	b6	b5	b4	b3	b2	b1	b0	
PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	$xx05
U	U	U	U	U	U	U	U	RESET

The port C latch register (PORTCL) allows alternate access to port C information. This register is used in conjunction with the strobed parallel I/O modes. Input data is latched into the PORTCL register on each selected edge on the STRA pin. The latched data is the level at the pins regardless of the operating mode selected. Reads of PORTCL return the contents of the port C input latch. Reads also act as part of an automatic flag clearing sequence in the input handshake modes of port C.

Writes to the PORTCL register are equivalent to writes to the PORTC register except the PORTCL writes are used as part of an automatic flag clearing sequence in the output handshake modes of port C. For more information on the port C strobed and handshake modes, see **I/O PORTS**. The contents of PORTCL are not affected by $\overline{\text{RESET}}$.

DATA DIRECTION REGISTER C (DDRC)

b7	b6	b5	b4	b3	b2	b1	b0	
DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	$xx07
0	0	0	0	0	0	0	0	RESET

The data direction register C (DDRC) is a read/write register used in conjunction with port C to specify the direction of data flow at each of the port C pins. A port C pin is an input if the corresponding bit in DDRC is zero. The pin is an output if the corresponding bit in DDRC is set to one. During reset, all bits in the DDRC are cleared to zero. The effects of DDRC are overridden in the three-state variation of the output handshake mode. For additional information, see **I/O PORTS, Output Handshake Protocol, Three-State Variation.**

HIGHEST PRIORITY INTERRUPT REGISTER (HPRIO)

b7	b6	b5	b4	b3	b2	b1	b0	
–	SMOD	–	IRV	–	–	–	–	$xx3C
0	–	0	–	0	0	0	0	RESET

NOTE

Reset condition of SMOD and IRV depend on initialization mode.

b7, b5, b3, b2, b1, b0—Not implemented

These bits are not implemented. Writes have no meaning or effect on them. Reads of these bits will always return a logic zero value.

b6, SMOD The SMOD (Special Test Mode) bit is a read only bit which reflects the operating mode of the peripheral as selected by the MODE input. The inverted state of MODE is latched in SMOD by the rising edge of $\overline{\text{RESET}}$. When SMOD equals zero (MODE equals one), the peripheral is operating in normal mode. When SMOD equals one (MODE equals zero), the special test mode is selected.

The special test mode may be exited under software control by writing SMOD from a one to a zero. However, the special test mode may not be reentered by writing the bit back to one. This SMOD bit becomes write-protected once written to zero. This implies that the normal operating mode can be entered either through a hardware reset or through software while the special test mode may only be entered through a hardware reset.

b4, IRV The IRV (Internal Read Visibility) control bit eliminates potential bus conflict problems when this device is used in conjunction with the MC68HC11. To allow a logic analyzer to monitor the internal bus activity of the MC68HC11, provisions have been made for the MPU to selectively drive the external data bus during internal reads as well as writes. The selection of this feature is controlled by the IRV bit.

The state following reset and the programming characteristics of the MC68HC24 IRV bit are the same as the MC68HC11 IRV bit. However, the functional characteristics are the opposite. The MC68HC24 IRV functions as follows:

Logic 0—Reads of the INIT and HPRIO registers will enable the multiplexed address/data buffers, placing the contents of the selected register on the bus.

Logic 1—Reads of the INIT and HPRIO registers do *not* enable the multiplexed address/data bus drivers.

This bit may be read at any time, although the multiplexed address/data bus will remain high-impedance during reads when IRV equals one. Only one write will be acknowledged and then only if SMOD equals one. The IRV bit is forced to zero (reads of HPRIO and INIT enabled) when SMOD is written from a one to a zero (entering normal mode). Reset clears this bit in the normal mode and sets this bit in the special test mode.

INIT (I/O MAPPING REGISTER)

b7	b6	b5	b4	b3	b2	b1	b0	
–	–	–	–	REG3	REG2	REG1	REG0	$xx3D
0	0	0	0	0	0	0	1	RESET

The INIT (I/O Mapping) register is a special purpose 8-bit register that is used (optionally) during initialization to change the default locations of the MC68HC24 internal registers in the MPU/MCU memory map. The lower four bits of the MC68HC24 INIT register are duplicates of the MC68HC11 INIT register. These four bits are used to specify the active state of the four high order address bits to the register address decoding logic. This register functions identically to the MC68HC11 INIT register with the following exceptions: 1) only the lower four bits are implemented, and 2) the protection mechanism is not time dependent.

The default starting address of the 64-byte internal register space is $1x00 (i.e., INIT is initialized to $01). Initialization software can move registers to any 4K boundary within the memory map. External decoding of A8 through A11 specifies where in the 4K block (on 256-byte boundaries) the 64-byte register space is located. As an example, assume that the initialization software wrote the value $09 to the INIT register and that $\overline{CS}$ was true when A8 through A11 were low. This would place the registers from $9000 through $903F in the memory map. Decoding A8 through A11 so that the chip is selected when all four address lines are low maps the MC68HC24 registers to the same address as the MC68HC11 registers.

The INIT register is special in that there is a write-protect mechanism associated with it. In the normal mode, the register may be written once at any time after reset. This *differs* from the operation of the MC68HC11 INIT register which becomes write protected after the first 64 E-clock cycles, whether or not a write to the register has occurred. After the first write, the MC68HC24 INIT register becomes write-protected and thereafter is a read-only register.

While in the special test mode (SMOD equals one), the protection mechanism is overridden and the INIT register may be written repeatedly as long as SMOD remains a one. When SMOD is written to a zero (to enter the normal operating mode), the write-protect mechanism is enabled. One additional write, regardless of the number of writes performed while in the special test mode, is allowed after entering normal operating mode. Writes to the upper four bits of the INIT register have no effect on the register contents and reads will always return zeros in the most significant bit positions.

SYSTEM CONFIGURATION

The MC68HC24 allows an end user to configure the peripheral to his specific MCU system through the use of hard wired options such as the mode select pin (MODE) and by the use of internal registers under software control. The following section describes those options which are fixed through hardware. Other configuration options, which can be changed dynamically, are discussed in the sections entitled **I/O PORTS** and **MODES OF OPERATION.**

MODE SELECTION

A dedicated mode select pin (MODE) determines which of two operating modes the MC68HC24 enters out of $\overline{\text{RESET}}$. Both modes properly emulate the action of Ports B and C of the MC68HC11. The modes are the normal and special test modes. Another dedicated pin (IOTEST) is used to test the output buffers.

The state of the mode select pin (MODE) is latched into the peripheral by the rising edge of $\overline{\text{RESET}}$ with the inverse of the latched value reflected in the SMOD bit of the HPRIO register. Normal mode is indicated by SMOD equals zero (MODE equals one). Special Test mode is indicated by SMOD equals one (MODE equals zero). The difference between these two modes is limited to the operation of the INIT and HPRIO registers.

The MODE input corresponds to the MODB input of the MC68HC11. In normal operation, this special test mode is not used, and the mode pin on both the MC68HC11 and the MC68HC24 can be tied to V_{DD}.

STATE AFTER RESET

When a low level is sensed on the $\overline{\text{RESET}}$ pin, the MC68HC24 enters the reset state. Most of the registers and control bits are forced to a specific state during reset and, if a user requires a different configuration, he must write the desired values into these registers in his initialization software. For detailed information about the options available, see **INTERNAL REGISTER DESCRIPTION.**

Note that $\overline{\text{RESET}}$ is synchronized to the system clock (E) before being used internally. For this reason, $\overline{\text{RESET}}$ must be held low for a minimum of two E-clock cycles to be recognized. Once recognized, the peripheral is initialized as described below.

Most of the configuration state after reset is independent of the selected operating mode. The STAF, STAI, and HNDS bits in the PIOC register are initialized to zeros so that no interrupt is pending or enabled and the simple strobed mode (rather than full handshake modes) of parallel I/O is selected. The CWOM bit is initialized to zero (Port C not operating in wired-OR mode). Port C is initialized as a general purpose, high-impedance input port (DDRC equals $00), STRA as an edge-sensitive strobe input, and the active edge is initially configured to detect rising edges (EGA bit set to one by $\overline{\text{RESET}}$). The STRB strobe output is initially a zero (INVB bit is initialized to one), while Port B is initialized with all outputs forced low.

The SMOD and IRV bits in the HPRIO register reflect the status of the MODE input at the rising edge of $\overline{\text{RESET}}$. Reset also deselects the chip and forces the multiplexed address/data bus to high impedance inputs.

MODES OF OPERATION

SPECIAL TEST MODE

The special test mode is selected with MODE equal to zero at the rising of edge of $\overline{\text{RESET}}$. Initialization into this mode loads HPRIO with $50 (SMOD and IRV equal one) and disables the INIT register write-protect mechanism.

While in special test mode (SMOD bit equals one), the INIT register write-protect mechanism is overridden and INIT remains writable as long as SMOD remains one. When SMOD is written to a zero (to enter the normal operating mode), the write-protect mechanism is enabled. One additional write is allowed after entering normal operating mode regardless of the number of writes perfomed while in the special test mode.

The reset state of IRV is one in the special test mode. An attempted read of either the INIT or HPRIO register with IRV equal to one will leave the data bus in a high impedance state with the output buffers disabled. If IRV equals zero, the data buffers are enabled and the contents of the selected register are placed on the data bus. The IRV bit is writable only one time while in the special test mode. Entering the normal mode forces the IRV bit to zero, enabling the data bus output buffers on reads of these two addresses. Table 1 summarizes the chip select options.

Table 1. MC68HC24 Chip Select Action Summary

$\overline{\text{CS}}$	IRV	Action Taken
0	0	Chip selected. HPRIO and INIT reads enabled.
0	1	Chip selected. HPRIO and INIT reads disabled.
1	X	Chip not selected.

NORMAL MODE

Normal mode is selected when the MODE input is at a logic high level at the rising edge of $\overline{\text{RESET}}$. The HPRIO register is initialized to $00 (SMOD and IRV equal zero). The INIT register write-protect mechanism is enabled, allowing only a single write to INIT. Reads of both the INIT and HPRIO register enable the output buffers, thus providing visibility into the contents of these registers. The HPRIO register is write-protected while in the normal mode. A reset sequence must be initiated to change the contents of this register.

NOTE

> A write to the INIT register must be included in the initialization software whether or not the registers are to be relocated. This write will ensure that an accidental write to register at a later time will not cause the registers to be remapped. THIS IS ONE OF THE FUNCTIONAL DIFFERENCES BETWEEN THE MC68HC11 PORTS AND THE MC68HC24 IMPLEMENTATION.

MC68HC11 AND MC68HC24 OPERATIONAL DIFFERENCES

INIT REGISTER WRITE-PROTECT MECHANISM

The MC68HC11 INIT register write-protect mechanism automatically disables writes to the INIT register 64 E clock cycles after the rising edge of $\overline{\text{RESET}}$. The MC68HC24 write-protect circuitry IS NOT TIME DEPENDENT. Only a write to the INIT register will disable further writes. Both the MC68HC11 and MC68HC24 INIT registers can be written repeatedly in the special test mode of operation (see **SPECIAL TEST MODE**) or once in the normal mode.

This difference dictates that the user should not rely on the timeout feature of the MC68HC11 to write-protect the INIT register if he plans to utilize the same software with the MC68HC24. Instead, a write to the INIT register should be done during initialization, even if the remapping feature is not going to be used.

STRA PULSE WIDTH

Due to differences in implementation technology, the MC68HC24 incorporates an additional level of synchronization (over the MC68HC11) on the STRA input. Under normal operating conditions, the end user will be unaware of this anomaly. Only systems which continually strobe new data into PORTCL are affected.

In order to allow the STRA signal to propagate through the internal feedback mechanism, a minimum delay of two E clock cycles between active edges has been specified. This delay should not concern most users, since the time required to acknowledge the receipt of data and to read the data is much greater that two cycles.

STRB SYNCHRONIZATION

The MC68HC11 synchronizes changes of port B, port C, and STRB data to an internal quadrature clock. This method of implementation makes internal buffer delays transparent to the end user. This internal clock is generated from the 4X clock, and as a result, cannot be duplicated by the MC68HC24. Port B and port C data are synchronized to the E clock and become valid t_{PWD} *after* the falling edge of E instead of a setup time before the falling edge of E.

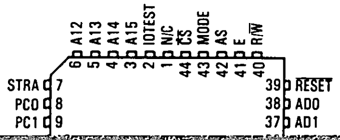
The most noticeable change involves STRB. The STRB signal is synchronized to the rising edge of E instead of the quadrature clock as in the MC68HC11. At slow clock rates (much less than 1 MHz), the delay between valid data on the port pins and the assertion of STRB could be considerable.

ORDERING INFORMATION

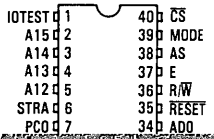
Package Type	Temperature Range	Order Number
Plastic P Suffix	−40 to +85°C −40 to +105°C −40 to +125°C	MC68HC24P MC68HC24VP MC68HC24MP
PLCC FN Suffix	−40 to +85°C −40 to +105°C −40 to +125°C	MC68HC24FN MC68HC24VFN MC68HC24MFN

PIN ASSIGNMENT

Quad


A12
A13
A14
A15
IOTEST
N/C
$\overline{CS}$
MODE
AS
E
R/$\overline{W}$
6
5
4
3
2
1
44
43
42
41
40
STRA 7
PC0 8
PC1 9
39 $\overline{RESET}$
38 AD0
37 AD1

Dual-in-Line


IOTEST 1
A15 2
A14 3
A13 4
A12 5
STRA 6
PC0 7
40 $\overline{CS}$
39 MODE
38 AS
37 E
36 R/$\overline{W}$
35 $\overline{RESET}$
34 AD0

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MCM68HC34

Advance Information

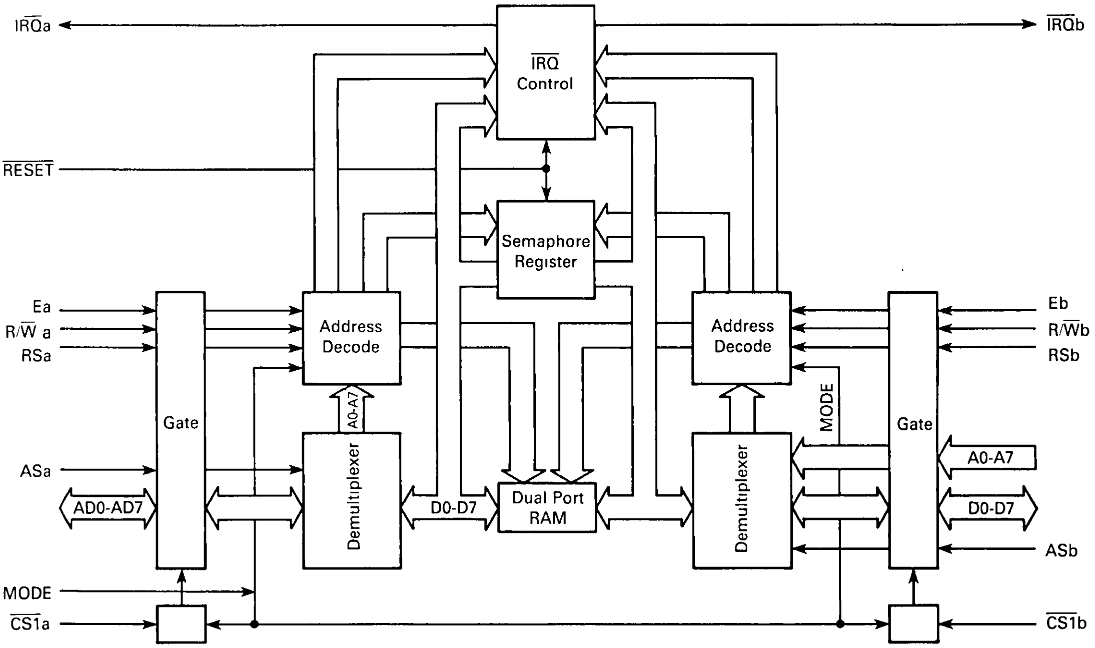
Dual-Port RAM Memory Unit

The MCM68HC34 is a dual-port RAM memory (DPM) unit which enables two processors, arbitrarily referred to a "A" and "B", operating on two separate buses to exchange data without interfering with devices on the other bus. It contains 256 bytes of dual-port RAM which is the medium actually used for the interchange of data.

The dual-port memory unit contains six semaphore registers that provide a means for controlling access to the dual-port RAM or any other shared resources. It also contains interrupt registers which provide a means for the processors to interrupt each other.

- High-Speed CMOS (HCMOS) Structure
- Six Read/Write Semaphore Registers
- 256 Bytes of Dual-Port RAM
- Eight Address Lines

FIGURE 1 — BLOCK DIAGRAM


IRQa
IRQ Control
IRQb
RESET
Semaphore Register
Ea
R/W a
RSa
Gate
Address Decode
A0-A7
Demultiplexer
ASa
AD0-AD7
MODE
CS1a
D0-D7
Dual Port RAM
Address Decode
MODE
Demultiplexer
Gate
Eb
R/Wb
RSb
A0-A7
D0-D7
ASb
CS1b

This document contains information on a new product. Specifications and information herein are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS

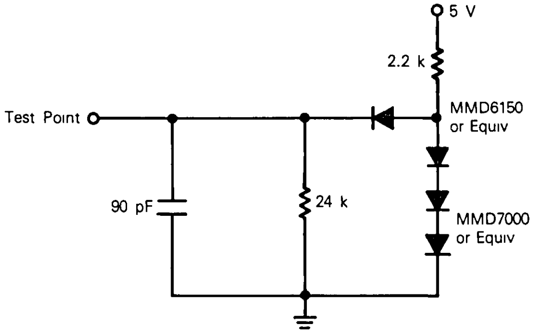
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to 7.0	V
Input Voltage All Inputs	V_{in}	$V_{SS}-0.3$ to $V_{CC}+0.5$	V
Operating Temperature	T_A	0 to 70	°C
Storage Temperature	T_{stg}	−55 to 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Unused inputs must be tied to an appropriate logic level (either V_{CC} or V_{SS}) to reduce leakage currents and increase reliablity.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Cerdip		60	
Plastic		100	
PLCC		60	

FIGURE 2 — BUS TIMING LOAD



DC ELECTRICAL CHARACTERISTCS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Input High Voltage (see Note 1)	V_{IH}	2.0	$V_{CC}+0.3$	V
Input Low Voltage (see Note 2)	V_{IL}	$V_{SS}-0.3$	0.8	V
Input Current (V_{in}=0 to V_{CC})	I_{in}	–	1.0	μA
Output Leakage Current	I_{OZ}	–	10.0	μA
Output High Voltage (I_{Load} = −100 μA) (I_{Load} = < 10.0 μA)	V_{OH}	2.4 $V_{CC}-0.1$	– –	V
Output Low Voltage (I_{Load} = 1.6 mA) (I_{Load} = < 10.0 μA)	V_{OL}	– –	0.4 0.1	V
Current Drain — Outputs Unloaded Operating — Ea, Eb = 1 MHz, Both Sides Active	I_{DD}	–	30	mA
Input Capacitance	C_{in}	–	10	pF
Output Capacitance (AD0-AD7 and D0-D7)	C_{out}	–	12	pF

NOTES.

1. Input high voltage as stated is for all inputs except MODE. In the case of MODE, input high voltage is tied to V_{CC}
2. Input low voltage as stated is for all inputs except MODE. In the case of MODE, input low voltage is tied to V_{SS} or is floating. If floating, the voltage will be internally pulled to V_{SS}

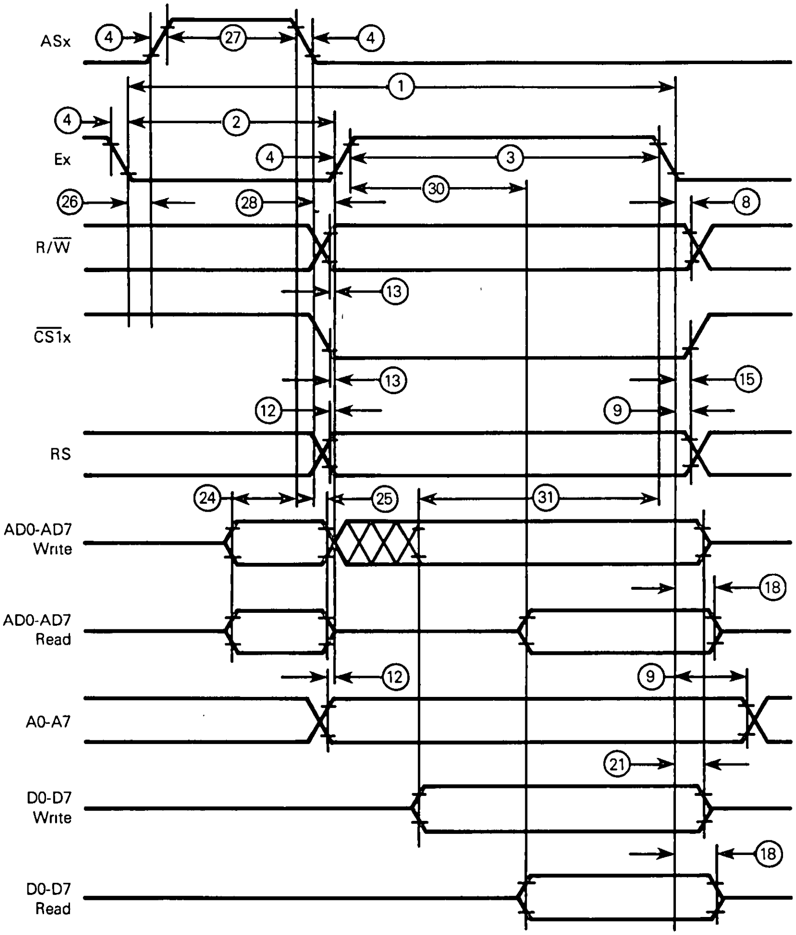
BUS TIMING (See Notes 1 and 2 and Figure 2)

Ident Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	800	–	ns
2	Pulse Width, E Low	PW_{EL}	300	–	ns
3	Pulse Width, E High	PW_{EH}	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	30	ns
8	Read/Write Hold Time	t_{RWH}	10	–	ns
9	Non-Multiplexed Address, RS Hold Time	t_{AH}	10	–	ns
12	Non-Multiplexed Address, RS Valid Time to Eb	t_{AV}	20	–	ns
13	R/$\overline{W}$, Chip Select Setup Time	t_{RWS}	20	–	ns
15	Chip Select Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	20	75	ns
21	Write Data Hold Time	t_{DHW}	10	–	ns
24	Address Setup Time for Latch	t_{ASL}	20	–	ns
25	Address Hold Time for Latch	t_{AHL}	20	–	ns
26	Delay Time E to AS Rise	t_{ASD}	60	–	ns
27	Pulse Width, AS High	PW_{ASH}	110	–	ns
28	Address Strobe to E Delay	t_{ASED}	20	–	ns
30	Read Data Delay Time	t_{DDR}	–	240	ns
31	Write Data Setup Time	t_{DSW}	100	–	ns

NOTES.

1. Timing numbers relative to one side only. No numbers are intended to be cross-referenced from one side to the other.
2. Measurement points shown for ac timing are 0 8 V and 2.0 V, unless otherwise specified.

BUS TIMING DIAGRAMS


ASx
Ex
R/W̄
C̄S̄1̄x
RS
AD0-AD7 Write
AD0-AD7 Read
A0-A7
D0-D7 Write
D0-D7 Read
①
②
③
④
⑧
⑨
⑫
⑬
⑮
⑱
㉑
㉔
㉕
㉖
㉗
㉘
㉚
㉛

SIGNAL DESCRIPTION

The following paragraphs contain a brief description of the input and output signals.

V_{CC} AND V_{SS}

These pins supply power to the DPM. V_{CC} is +5 volts ± 5% and V_{SS} is 0 volts or ground.

E CLOCK INPUTS (Ea AND Eb)

These are the input clocks from the respective processors and are positive during the latter portion of the bus cycle.

REGISTER SELECT INPUTS (RSa AND RSb)

These inputs function as register select inputs. A high on the RSa for side A or RSb for side B input allows selection of the semaphore and interrupt registers respectively for side A and side B by the lower three address bits. A low on RSa or RSb selects 256 bytes of RAM from side A or side B respectively.

CHIP SELECT INPUTS ($\overline{\mathrm{CS1}}$a AND $\overline{\mathrm{CS1}}$b)

These inputs function as chip select inputs for their respective sides. $\overline{\mathrm{CS1}}$a must be low to select side A and $\overline{\mathrm{CS1}}$b must be low to select side B. If $\overline{\mathrm{CS1}}$a is high, side A is deselected. If $\overline{\mathrm{CS1}}$b is high, side B is deselected.

MODE SELECT (MODE)

In normal operation, this pin should always be connected to V_{CC} (MODE = 1). Each side has three states controlled by RSa and $\overline{\mathrm{CS1}}$a for side A and RSb and $\overline{\mathrm{CS1}}$b for side B.

If $\overline{\mathrm{CS1}}$a is high, side A cannot be accessed. If $\overline{\mathrm{CS1}}$a is low, side A accesses either 256 bytes of RAM or the six semaphore registers and the two interrupt registers depending on the level of RSa. If RSa is low, 256 bytes of RAM are accessed and if RSa is high, the six semaphores and two interrupt registers are accessed.

The six semaphore and two interrupt registers are redundantly mapped in the 256 byte mode. That is, only the low order three bits select one of eight registers and the upper five bits of address are not decoded. Refer to Table 1.

TABLE 1 — SIDE A CONTROL SIGNAL OPERATION

Mode	$\overline{\mathrm{CS1}}$a	RSa	Operation
1	0	0	Access 256 Byte RAM Side A
1	0	1	Access Semaphore/$\overline{\mathrm{IRQ}}$ Side A on Lower Three Bits of Address
1	1	X	Side A Not Selected

The three states for side B in the 256 byte mode are controlled in the manner as side A using RSb and $\overline{\mathrm{CS1}}$b except that side B uses separated address and data inputs. Refer to Table 2.

TABLE 2 — SIDE B CONTROL SIGNAL OPERATION

Mode	$\overline{\mathrm{CS1}}$b	RSb	Operation
1	0	0	Access 256 Byte RAM Side B
1	0	1	Access Semaphore/$\overline{\mathrm{IRQ}}$ Side B on Lower Three Bits of Address
1	1	X	Side B Not Selected

INTERRUPT REQUEST OUTPUTS ($\overline{\mathrm{IRQ}}$a AND $\overline{\mathrm{IRQ}}$b)

These pins are active low open-drain outputs. A write to address F9 from one side asserts an interrupt, if not masked on the other side. A write to address F9 sets this pin low.

B SIDE ADDRESS BUS INPUTS (A0-A7) AND B SIDE BIDIRECTIONAL DATA BUS (D0-D7)

When the B side is run from a multiplexed bus processor, the B side address pins are connected to the B side data pins, respectively (A0 to D0, A1 to D1, etc.).

SYSTEM RESET INPUT ($\overline{\mathrm{RESET}}$)

A low level on this input causes the semaphore registers to be set to the states shown in Table 5 under **SEMAPHORE REGISTERS** and clears both bits of both $\overline{\mathrm{IRQ}}$ registers to zeros. The RAM data is unaffected by $\overline{\mathrm{RESET}}$.

ADDRESS STROBE INPUTS (ASa AND ASb)

The ASa input demultiplexes the eight low order address lines from the data lines on the A side. The falling edge of ASa latches the A side address within the DPM. The ASb input is used in the same manner when the B side is connected to a multiplexed bus. It must be connected to a high level when the B side is connected to a non-multiplexed bus.

A SIDE MULTIPLEXED ADDRESS/ BIDIRECTIONAL DATA BUS (AD0-AD7)

The A side can only be used with a multiplexed address/data bus. The A side addresses are on these lines during the time ASa is high. The lines are used as bidirectional data lines during the time Ea is high.

DUAL-PORT RAM

The dual-port memory unit contains 256 bytes of dual-port RAM that is accessed from either processor. It is selected in either case by eight address lines, register select, and chip select inputs. The direction of data transfer is controlled by the respective read/write (R/$\overline{\mathrm{W}}$a or R/$\overline{\mathrm{W}}$b) line. The dual-port RAM enables the processors to exchange data without interfering with devices on the other bus.

Simultaneous accesses by both sides of different locations of dual-port RAM will cause no ambiguities. Simultaneous reads by both sides of the same dual-port RAM location gives the proper data to both sides. On a simultaneous write and read of the same location, the data written is put into RAM but the data read is undefined. Simultaneous writes to the same RAM location result in undefined data being stored. Thus, simultaneous writes and simultaneous write and read to the same location should be avoided. The semaphore registers provide a tool for determining when the shared RAM is available.

SEMAPHORE REGISTERS

The dual-port memory unit contains six read/write semaphore registers. Only two bits of each register are used. Bit 7 is the semaphore (SEM) bit and bit 6 is the ownership (OWN) bit. The remaining six bits will read all zeros.

Each semaphore register is able to arbitrate simultaneous accesses to it. The semaphore register bits provide a mechanism for controlling accesses to the shared RAM but there are no hardware controls of the dual-port RAM by the semaphore registers.

Table 3 is the truth table for when a semaphore register is accessed by one of the processors. When a semaphore register is written, the actual data written is disregarded but the SEM bit is set to zero. When the register is read, the resulting SEM bit is one (for the next read). The data obtained from the read is interpreted as: SEM bit equals zero — resource available, SEM bit equals one — resource not available.

TABLE 3 — ONE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	$R/\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	$\overline{W}$	–	0
1	$\overline{W}$	–	0

*0 = Resource Available
1 = Resource Not Available

Table 4 shows the truth table if both processors read or read and write the same semaphore register at the same time. The A processor always reads the actual SEM bit. The B processor reads the SEM bit except during the simultaneous read of a clear SEM bit. This ensures that during a simultaneous read, only the A processor reads a clear SEM bit and therefore has priority to the shared RAM.

TABLE 4 — SIMULTANEOUS ACCESS OF OF SEMAPHORE REGISTER TRUTH TABLE

Original	A Processor		B Processor		Resulting
SEM Bit	$R/\overline{W}$	Data Read	$R/\overline{W}$	Data Read	SEM Bit
0	R	0*	R	1*	1
1	R	1*	$\overline{W}$	–	0
1	$\overline{W}$	–	R	1*	0
1	R	1*	R	1*	1

*0 = Resource Available
1 = Resource Not Available

The ownership bit is a read-only bit that indicates which processor last set the SEM bit. The OWN bit is set to a one whenever the SEM bit is set from zero to one The OWN bit as read by one processor is the complement of the bit read by the other processor.

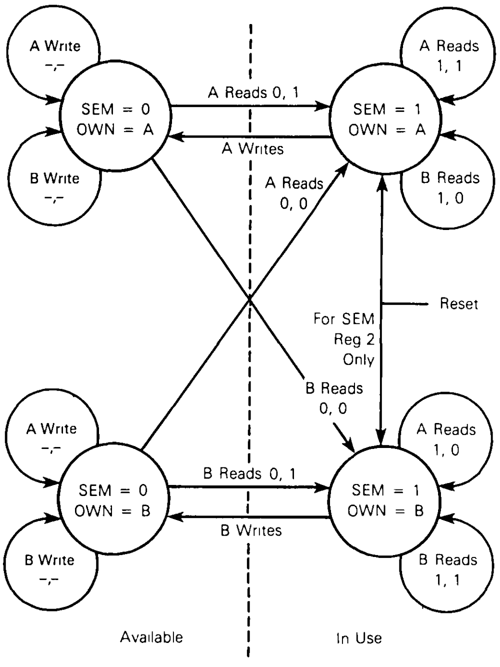
The reset state of the semaphore registers is defined in Table 5. The A processor owns all of the semaphore registers except the second semaphore register which is owned by the B processor.

TABLE 5 — RESET STATE OF SEMAPHORE REGISTERS

Semaphore Register	A Processor		B Processor	
Number	SEM Bit	OWN Bit	SEM Bit	OWN Bit
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

A state diagram for a semaphore register is shown in Figure 3.

FIGURE 3 — STATE DIAGRAM FOR SEMAPHORE REGISTER



NOTES
1 Writes to a semaphore register are valid only if SEM = 1 and OWN = 1
2 When A and B simultaneously read a semaphore register, the hardware handles it as a read by A followed by a read by B

INTERRUPT REGISTERS

The dual-port memory unit contains two addressable locations at F8 and F9 on both sides that control the interrupt ($\overline{IRQ}$) operation between the processors. Although there is only one hardware register for each side, for purposes of explanation the register accessed at location F8 is referred to as the IRQX status register and the register accessed at location F9 is referred to as the IRQX control register (refer to Table 6). The registers each consisting of two bits have identical bit arrangements. Bit 6 is the enable bit and bit 7 is the flag bit. The other six bits are not used and always read as zero. When $\overline{\text{RESET}}$ is asserted, both bits are cleared to zero.

Table 7 summarizes the bits involved when reading or writing to the status or control registers at F8 or F9. The enable bits on either side (A or B) track the data that is written into the status register from that side. Writes to the control register do not alter data. The actual data written is disregarded but the action sets the flag bit in the other side's register and asserts an interrupt signal if enabled.

The following describes how the B side interrupt is asserted from the A side. The A side interrupt is controlled in a similar manner.

When the enable bit in the IRQb status register is set (bit 6 = 1), a write to IRQa control register sets the flag bit in the IRQb status register (bit 7 = 1) and causes an interrupt on the B side by setting the IRQb pin low. Reading the IRQb status register reads the state of the B side enable and flag bits. Reading the IRQb control register also reads the enable and flag bits but in addition, clears the B side flag bit (bit 7 = 0) and clears the B side interrupt by removing the low condition on the IRQb pin.

The enable bit in the IRQb status register (bit 6) is changed by writing the proper data to bit 6 of the IRQb status register. If the B side enable bit is zero, interrupts are prevented on the B side. However, a write to the IRQa control register still sets the B side flag bit.

INTERNAL REGISTER ADDRESSES

Table 8 shows the address of the RAM, $\overline{\text{IRQ}}$, and semaphore registers. The addresses to these registers are the same whether accessed from the A or B side. The address and data buses are multiplexed on the A side. The B side has separate address and data buses. The B side can be used on a multiplexed bus by connecting the corresponding address and data bit pins together (A0 to D0, A1 to D1, etc.) and using the B side address strobe input pin.

TABLE 6 — IRQ REGISTERS

Location	Register Name	Bit 7	Bit 6	Bits 5 to 0
A Side F8	IRQa Status	Flag	Enable	Not Used
A Side F9	IRQa Control	Flag	Enable	Not Used
B Side F8	IRQb Status	Flag	Enable	Not Used
B Side F9	IRQb Control	Flag	Enable	Not Used

TABLE 7 — INTERRUPT OPERATION

Operation	Action Taken
A Reads IRQa Status at F8	Read EA and FA
A Writes IRQa Status at F8	Writes to EA
A Reads IRQa Control at F9	Read EA and FA; Clear FA
A Writes IRQa Control at F9	Set FB; Assert IRQB if Enabled
B Reads IRQb Status at F8	Read EB and FB
B Writes IRQb Status at F8	Writes to EB
B Reads IRb Control at F9	Read EB and FB; Clear FB
B Writes IRQb Control at F9	Set FA; Assert IRQA if Enabled

F8 and F9 are Address Locations
EA and FA are A Side Enable and Flag Bits
EB and FB are B Side Enable and Flag Bits

TABLE 8 — REGISTER LOCATIONS

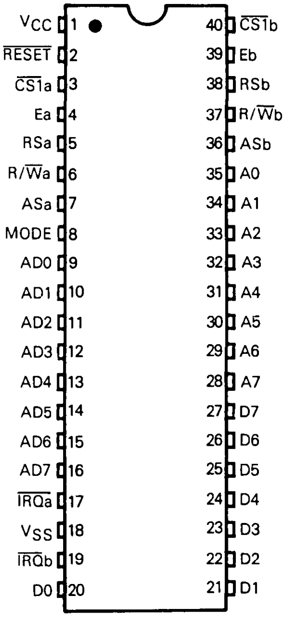
RS	Address	Register Name
0	00-FF	Dual Ported RAM
1	00-07	IRQ and Semaphore
1	08-0F	IRQ and Semaphore
1	10-17	IRQ and Semaphore
1	18-1F	IRQ and Semaphore
	•	
1	•	IRQ and Semaphore
	•	
1	E0-E7	IRQ and Semaphore
1	E8-EF	IRQ and Semaphore
1	F0-F7	IRQ and Semaphore
1	F8-FF	IRQ and Semaphore

Where.
X is 0 through F of the upper four bits of the address (note that only the lower three bits of the address are decoded):
X0 and X8 IRQa or IRQb Status
X1 and X9 IRQa or IRQb Control
X2 and XA Semaphore 1
X3 and XB Semaphore 2
X4 and XC Semaphore 3
X5 and XD Semaphore 4
X6 and XE Semaphore 5
X7 and XF Semaphore 6

ORDERING INFORMATION ($T_A = 0°$ to $70°C$)

Package Type	Order Number
Cerdip S Suffix	MCM68HC34S
Plastic P Suffix	MCM68HC34P
PLCC FN Suffix	MCM68HC34FN

PIN ASSIGNMENT


VCC 1
RESET 2
CS1a 3
Ea 4
RSa 5
R/Wa 6
ASa 7
MODE 8
AD0 9
AD1 10
AD2 11
AD3 12
AD4 13
AD5 14
AD6 15
AD7 16
IRQa 17
VSS 18
IRQb 19
D0 20
40 CS1b
39 Eb
38 RSb
37 R/Wb
36 ASb
35 A0
34 A1
33 A2
32 A3
31 A4
30 A5
29 A6
28 A7
27 D7
26 D6
25 D5
24 D4
23 D3
22 D2
21 D1

MC6840

Programmable Timer Module (PTM)

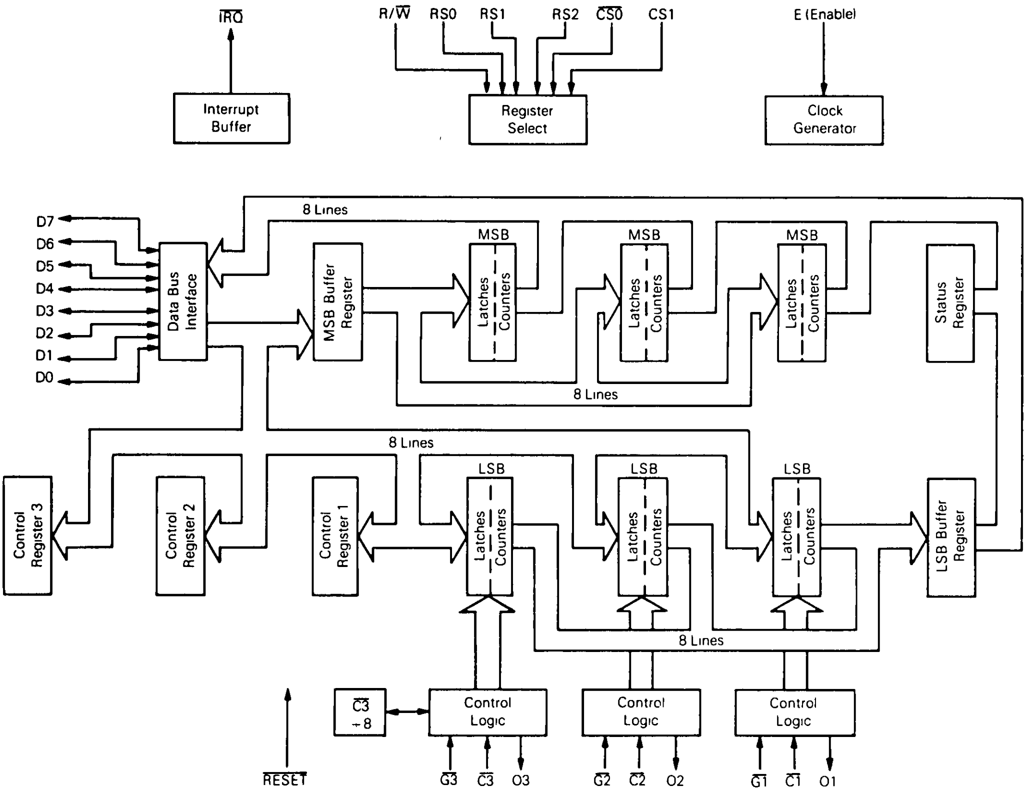
The MC6840 is a programmable subsystem component of the M6800 Family designed to provide variable system time intervals.

The MC6840 has three 16-bit binary counters, three corresponding control registers, and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring, and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

- Operates from a Single 5-Volt Power Supply
- Fully TTL Compatible
- Single System Clock Required (Enable)
- Selectable Prescaler on Timer 3 Capable of 4 MHz for the MC6840, 6 MHz for the MC68A40 and 8 MHz for the MC68B40
- Programmable Interrupts ($\overline{\text{IRQ}}$) Output to MPU
- Readable Down Counter Indicates Counts to Go Until Time-Out
- Selectable Gating for Frequency or Pulse-Width Comparison
- $\overline{\text{RESET}}$ Input
- Three Asynchronous External Clock and Gate/Trigger Inputs Internally Synchronized
- Three Maskable Outputs

This document contains information on a new product. Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM


IRQ
R/W
RS0
RS1
RS2
CS0
CS1
E (Enable)
Interrupt Buffer
Register Select
Clock Generator
D7
D6
D5
D4
D3
D2
D1
D0
Data Bus Interface
8 Lines
MSB Buffer Register
MSB
Latches
Counters
Status Register
8 Lines
8 Lines
Control Register 3
Control Register 2
Control Register 1
LSB
LSB Buffer Register
8 Lines
C3 ÷8
Control Logic
RESET
G3
C3
O3
G2
C2
O2
G1
C1
O1

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range − T_L to T_H MC6840, MC68A40, MC68B40 MC6840C, MC68A40C	T_A	0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Cerdip		65	
Plastic		100	

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage						
($I_{Load} = -205\ \mu A$)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
($I_{Load} = -200\ \mu A$)	Other Outputs		$V_{SS}+2.4$	–	–	
Output Low Voltage						
(I_{Load} = 1.6 mA)	$\overline{\text{IRQ}}$, D0-D7	V_{OL}	–	–	$V_{SS}+0.4$	V
(I_{Load} = 3.2 mA)	O1-O3		–	–	$V_{SS}+0.4$	
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{\text{IRQ}}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	470	700	mW
Input Capacitance						
(V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Others		–	–	7.5	
Output Capacitance						
(V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{\text{IRQ}}$	C_{out}	–	–	5.0	pF
	O1, O2, O3		–	–	10	

AC OPERATING CHARACTERISTICS (See Figures 2-7)

Characteristic	Symbol	MC6840		MC68A40		MC68B40		Unit
		Min	Max	Min	Max	Min	Max	
Input Rise and Fall Times (Figures 4 and 5) $\overline{\text{C}}$, $\overline{\text{G}}$, and $\overline{\text{RESET}}$	t_r, t_f	–	1.0*	–	0.666*	–	0.500*	μs
Input Pulse Width Low (Figure 2) (Asynchronous Input) $\overline{\text{C}}$, $\overline{\text{G}}$, and $\overline{\text{RESET}}$	PW_L	$t_{cycE}+t_{su}+t_{hd}$	–	$t_{cycE}+t_{su}+t_{hd}$	–	$t_{cycE}+t_{su}+t_{hd}$	–	ns
Input Pulse Width High (Figure 3) (Asynchronous Input) $\overline{\text{C}}$, $\overline{\text{G}}$	PW_H	$t_{cycE}+t_{su}+t_{hd}$	–	$t_{cycE}+t_{su}+t_{hd}$	–	$t_{cycE}+t_{su}+t_{hd}$	–	ns
Input Setup Time (Figure 4) (Synchronous Input) $\overline{\text{C}}$, $\overline{\text{G}}$, and $\overline{\text{RESET}}$	t_{su}	200	–	120	–	75	–	ns
Input Hold Time (Figure 4) (Synchronous Input) $\overline{\text{C}}$, $\overline{\text{G}}$, and $\overline{\text{RESET}}$	t_{hd}	50	–	50	–	50	–	ns
Input Synchronization Time (Figure 7) $\overline{\text{C3}}$ (÷8 Prescaler Mode Only)	t_{sync}	250	–	200	–	175	–	ns
Input Pulse Width $\overline{\text{C3}}$ (÷8 Prescaler Mode Only)	PW_L, PW_H	120	–	80	–	60	–	ns
Output Delay, O1-O3 (Figure 5)								
(V_{OH} = 2.4 V, Load B) TTL	t_{co}	–	700	–	460	–	340	ns
(V_{OH} = 2.4 V, Load D) MOS	t_{cm}	–	450	–	450	–	340	ns
($V_{OH} = 0.7\ V_{DD}$, Load D) CMOS	t_{cmos}	–	2.0	–	1.35	–	1.0	μs
Interrupt Release Time (Figure 6)	t_{IR}	–	1.2	–	0.9	–	0.7	μs

*t_r and $t_f \leq t_{cycE}$

BUS TIMING CHARACTERISTICS (See Notes 1, 2, and 3)

Ident. Number	Characteristic	Symbol	MC6840		MC68A40		MC68B40		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (High Impedance)

NOTES.
1. Not all signals are applicable to every part
2. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FIGURE 1 — BUS TIMING

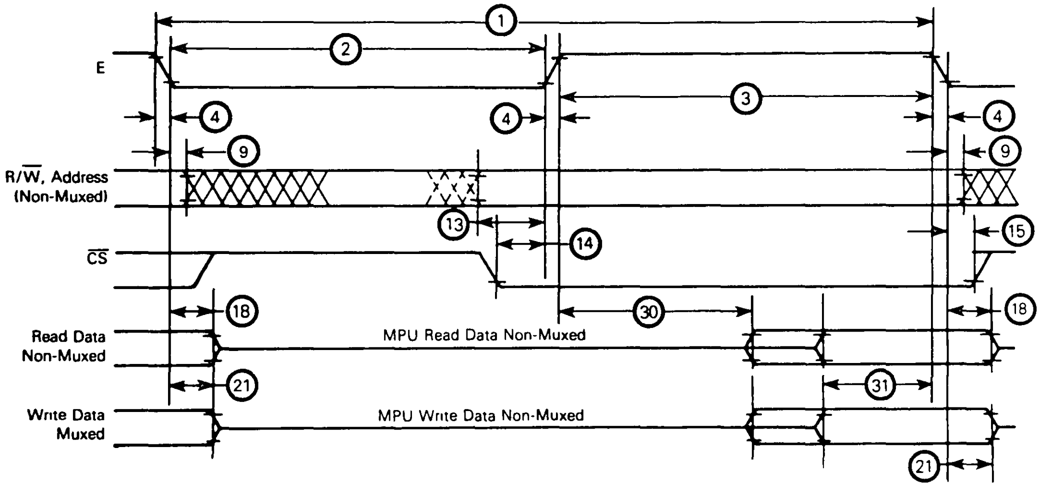

①
②
③
E
④
④
④
⑨
⑨
R/W, Address (Non-Muxed)
⑬
⑭
⑮
CS
⑱
㉚
⑱
Read Data Non-Muxed
MPU Read Data Non-Muxed
㉑
㉛
Write Data Muxed
MPU Write Data Non-Muxed
㉑

FIGURE 2 — INPUT PULSE WIDTH LOW

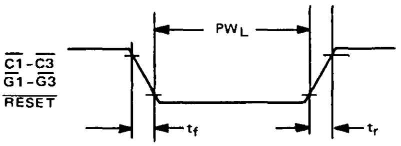

PW_L
C1-C3
G1-G3
RESET
t_f
t_r

FIGURE 3 — INPUT PULSE WIDTH HIGH

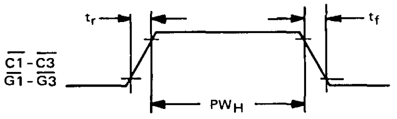

t_r
t_f
C1-C3
G1-G3
PW_H

FIGURE 4 — INPUT SETUP AND HOLD TIMES

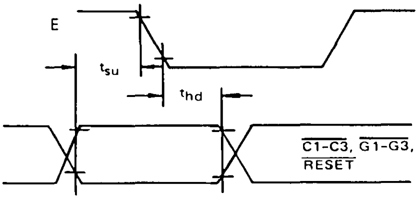

E
t_{su}
t_{hd}
$\overline{C1-C3}$, $\overline{G1-G3}$, $\overline{RESET}$

FIGURE 5 — OUTPUT DELAY

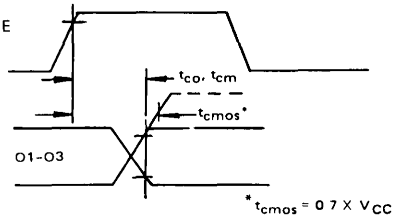

E
t_{co}, t_{cm}
t_{cmos}*
O1-O3
*t_{cmos} = 0.7 X V_{CC}

FIGURE 6 — IRQ RELEASE TIME

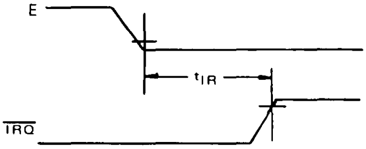

E
t_{IR}
$\overline{IRQ}$

FIGURE 7 — $\overline{C3}$ INPUT SYNCHRONIZATION TIME (÷8 PRESCALER MODE ONLY)

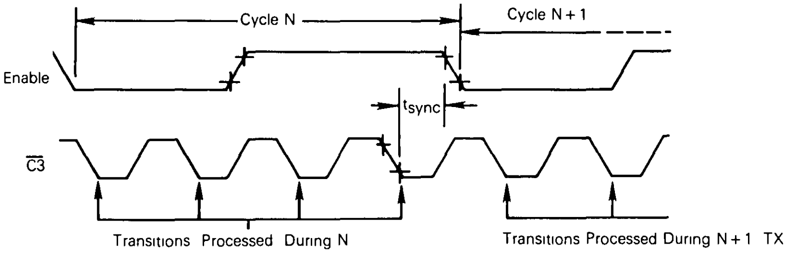
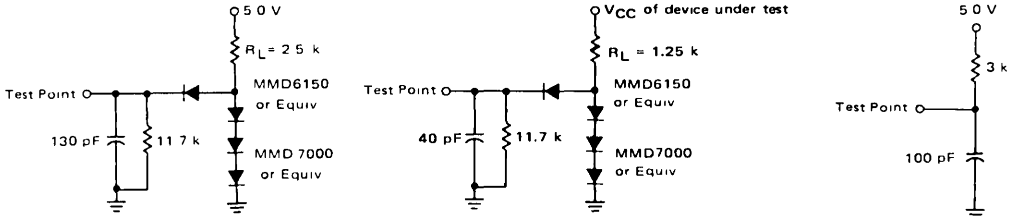

Cycle N
Cycle N + 1
Enable
t_{sync}
$\overline{C3}$
Transitions Processed During N
Transitions Processed During N + 1 TX

FIGURE 8 — BUS TIMING TEST LOADS

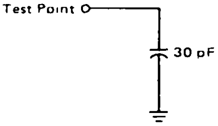

5.0 V
R_L = 2.5 k
Test Point
MMD6150 or Equiv
130 pF
11.7 k
MMD 7000 or Equiv
V_{CC} of device under test
R_L = 1.25 k
Test Point
MMD6150 or Equiv
40 pF
11.7 k
MMD7000 or Equiv
5.0 V
3 k
Test Point
100 pF

Load A
(D0–D7)

Load B
(O1, O2, O3)
(TTL Load)

Load C
($\overline{IRQ}$ Only)

Load D
(O1, O2, O3)
(CMOS Load)
(MOS)


Test Point
30 pF

NOTE Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

DEVICE OPERATION

The MC6840 is part of the M6800 microprocessor family and is fully bus compatible with M6800 systems. The three timers in the MC6840 operate independently and in several distinct modes to fit a wide variety of measurement and synthesis applications.

The MC6840 is an integrated set of three distinct counter/timers It consists of three 16-bit data latches, three 16-bit counters (clocked independently), and the comparison and enable circuitry necessary to implement various measurement and synthesis functions In addition, it contains interrupt drivers to alert the processor that a particular function has been completed.

In a typical application, a timer will be loaded by first storing two bytes of data into an associated Counter Latch. This data is then transferred into the counter via a Counter Initialization cycle. If the counter is enabled, the counter decrements on each subsequent clock period which may be an external clock, or Enable (E) until one of several predetermined conditions causes it to halt or recycle. The timers are thus programmable, cyclic in nature, controllable by external inputs or the MPU program, and accessible by the MPU at any time.

BUS INTERFACE

The Programmable Timer Module (PTM) interfaces to the M6800 Bus with an 8-bit bidirectional data bus, two Chip Select lines, a Read/Write line, a clock (Enable) line, and Interrupt Request line, an external Reset line, and three Register select lines. VMA should be utilized in conjunction with an MPU address line into a Chip Select of the PTM when using the MC6800/6802/6808

BIDIRECTIONAL DATA (D0-D7) — The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and PTM. The data bus output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a PTM read operation (Read/Write and Enable lines high and PTM Chip Selects activated)

CHIP SELECT ($\overline{\text{CS0}}$, CS1) — These two signals are used to activate the Data Bus interface and allow transfer of data from the PTM. With $\overline{\text{CS0}}$=0 and CS1=1, the device is selected and data transfer will occur

READ/WRITE (R/$\overline{\text{W}}$) — This signal is generated by the MPU to control the direction of data transfer on the Data Bus. With the PTM selected, a low state on the PTM R/$\overline{\text{W}}$ line enables the input buffers and data is transferred from the MPU to the PTM on the trailing edge of the E (Enable) clock Alternately, (under the same conditions) R/$\overline{\text{W}}$=1 and Enable high allows data in the PTM to be read by the MPU

ENABLE (E CLOCK) — The E clock signal synchronizes data transfer between the MPU and the PTM. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the PTM.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$) — The active low Interrupt Request signal is normally tied directly (or through priority interrupt circuitry) to the $\overline{\text{IRQ}}$ input of the MPU. This is an "open drain" output (no load device on the chip) which permits other similar interrupt request lines to be tied together in a wire-OR configuration

The $\overline{\text{IRQ}}$ line is activated if, and only if, the Composite Interrupt Flag (Bit 7 of the Internal Status Register) is asserted The conditions under which the $\overline{\text{IRQ}}$ line is activated are discussed in conjunction with the Status Register.

$\overline{\text{RESET}}$ — A low level at this input is clocked into the PTM by the E (Enable) input Two Enable pulses are required to synchronize and process the signal The PTM then recognizes the active "low" or inactive "high" on the third Enable pulse If the $\overline{\text{RESET}}$ signal is asynchronous, an additional Enable period is required if setup times are not met The $\overline{\text{RESET}}$ input must be stable High/Low for the minimum time stated in the AC Operating Characteristics

Recognition of a low level at this input by the PTM causes the following action to occur

a All counter latches are preset to their maximum count values

b All Control Register bits are cleared with the exception of CR10 (internal reset bit) which is set

c. All counters are preset to the contents of the latches

d. All counter outputs are reset and all counter clocks are disabled

e. All Status Register bits (interrupt flags) are cleared

REGISTER SELECT LINES (RS0, RS1, RS2) — These inputs are used in conjunction with the R/$\overline{\text{W}}$ line to select the internal registers, counters and latches as shown in Table 1

NOTE

The PTM is accessed via MPU Load and Store operations in much the same manner as a memory device. The instructions available with the M6800 family of MPUs which perform read-modify-write operations on memory should not be used when the PTM is accessed. These instructions actually fetch a byte from memory, perform an operation, then restore it to the same address location. Since the PTM uses the R/$\overline{\text{W}}$ line as an additional register select input, the modified data will not be restored to the same register if these instructions are used.

CONTROL REGISTER

Each timer in the MC6840 has a corresponding write-only Control Register Control Register #2 has a unique address space (RS0=1, RS=0, RS2=0) and therefore may be written into at any time The remaining Control Registers (#1 and #3) share the Address Space selected by a logic zero on all Register Select inputs

CR20 — The least significant bit of Control Register #2 (CR20) is used as an additional addressing bit for Control Registers #1 and #3. Thus, with all Register selects and R/$\overline{\text{W}}$ inputs at logic zero, Control Register #1 will be written into if CR20 is a logic one. Under the same conditions, Control Register #3 can also be written into after a $\overline{\text{RESET}}$ low condition has occurred, since all control register bits (except CR10) are cleared. Therefore, one may write in the sequence CR3, CR2, CR1

TABLE 1 — REGISTER SELECTION

Register Select Inputs			Operations	
RS2	RS1	RS0	R/$\overline{W}$ = 0	R/$\overline{W}$ = 1
0	0	0	CR20 = 0 Write Control Register #3 CR20 = 1 Write Control Register #1	No Operation
0	0	1	Write Control Register #2	Read Status Register
0	1	0	Write MSB Buffer Register	Read Timer #1 Counter
0	1	1	Write Timer #1 Latches	Read LSB Buffer Register
1	0	0	Write MSB Buffer Register	Read Timer #2 Counter
1	0	1	Write Timer #2 Latches	Read LSB Buffer Register
1	1	0	Write MSB Buffer Register	Read Timer #3 Counter
1	1	1	Write Timer #3 Latches	Read LSB Buffer Register

CR10 — The least significant bit of Control Register #1 is used as an Internal Reset bit. When this bit is a logic zero, all timers are allowed to operate in the modes prescribed by the remaining bits of the control registers Writing a "one" into CR10 causes all counters to be preset with the contents of the corresponding counter latches, all counter clocks to be disabled, and the timer outputs and interrupt flags (Status Register) to be reset. Counter Latches and Control Registers are undisturbed by an Internal Reset and may be written into regardless of the state of CR10

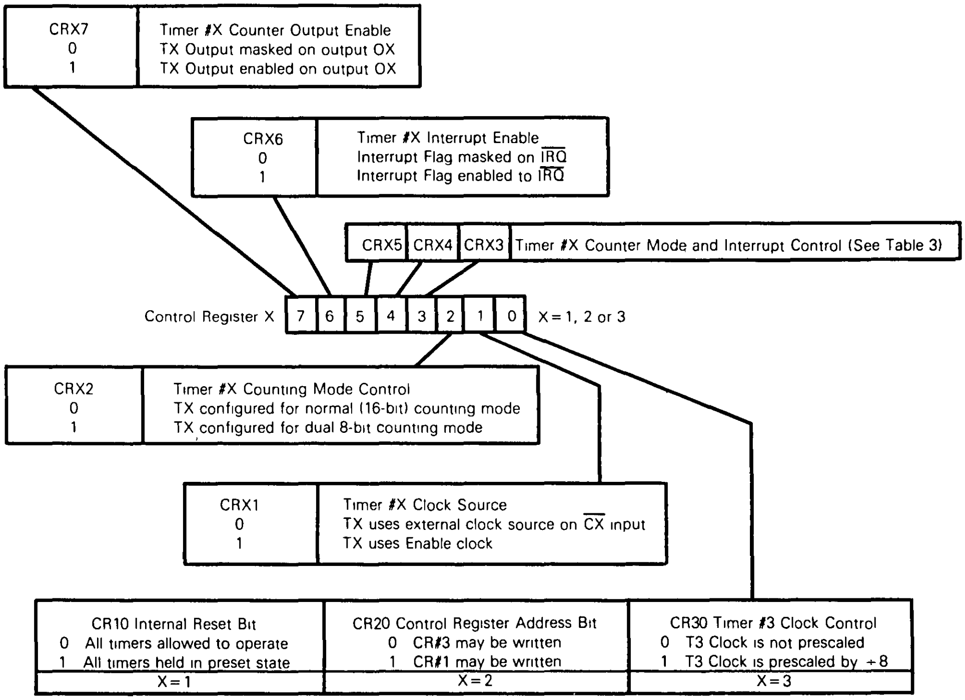
The least signifcant bit of Control Register #3 is used as a selector for a ÷8 prescaler which is available with Timer #3 only The prescaler, if selected, is effectively placed between the clock input circuitry and the input to Counter #3. It can therefore be used with either the internal clock (Enable) or an external clock source

NOTE

When initializing Timer 3 into the divide-by-eight mode on consecutive E-cycles (i.e., with DMA), Control Register 3 must be initialized before Timer Latch #3 to insure proper timer initialization.

CR30 — The functions depicted in the foregoing discussions are tabulated in Table 2 for ease of reference.

TABLE 2 — CONTROL REGISTER BITS


CRX7
0
1
Timer #X Counter Output Enable
TX Output masked on output OX
TX Output enabled on output OX
CRX6
0
1
Timer #X Interrupt Enable
Interrupt Flag masked on $\overline{IRQ}$
Interrupt Flag enabled to $\overline{IRQ}$
CRX5
CRX4
CRX3
Timer #X Counter Mode and Interrupt Control (See Table 3)
Control Register X
7
6
5
4
3
2
1
0
X = 1, 2 or 3
CRX2
0
1
Timer #X Counting Mode Control
TX configured for normal (16-bit) counting mode
TX configured for dual 8-bit counting mode
CRX1
0
1
Timer #X Clock Source
TX uses external clock source on $\overline{CX}$ input
TX uses Enable clock
CR10 Internal Reset Bit
0 All timers allowed to operate
1 All timers held in preset state
X=1
CR20 Control Register Address Bit
0 CR#3 may be written
1 CR#1 may be written
X=2
CR30 Timer #3 Clock Control
0 T3 Clock is not prescaled
1 T3 Clock is prescaled by ÷8
X=3

Control Register Bits CR10, CR20, and CR30 are unique in that each selects a different function. The remaining bits (1 through 7) of each Control Register select common functions, with a particular Control Register affecting only its corresponding timer.

CRX1 – Bit 1 of Control Register #1 (CR11) selects whether an internal or external clock source is to be used with Timer #1. Similarly, CR21 selects the clock source for Timer #2, and CR31 performs this function for Timer #3 The function of each bit of Control Register "X" can therefore be defined as shown in the remaining section of Table 2.

CRX2 – Control Register Bit 2 selects whether the binary information contained in the Counter Latches (and subsequently loaded into the counter) is to be treated as a single 16-bit word or two 8-bit bytes. In the single 16-bit Counter Mode (CRX2=0) the counter will decrement to zero after N+1 enabled (G=0) clock periods, where N is defined as the 16-bit number in the Counter Latches. With CRX2=1, a similar Time Out will occur after $(L+1)\bullet(M+1)$ enabled clock periods, where L and M, respectively, refer to the LSB and MSB bytes in the Counter Latches.

CRX3-CRX7 – Control Register Bits 3, 4, and 5 are explained in detail in the Timer Operating Mode section. Bit 6 is an interrupt mask bit which will be explained more fully in conjunction with the Status Register, and bit 7 is used to enable the corresponding Timer Output. A summary of the control register programming modes is shown in Table 3.

STATUS REGISTER/INTERRUPT FLAGS

The MC6840 has an internal Read-Only Status Register which contains four Interrupt Flags. (The remaining four bits of the register are not used, and defaults to zeros when being read.) Bits 0, 1, and 2 are assigned to Timers 1, 2, and 3, respectively, as individual flag bits, while Bit 7 is a Composite Interrupt Flag. This flag bit will be asserted if any of the individual flag bits is set while Bit 6 of the corresponding Control Register is at a logic one. The conditions for asserting the composite Interrupt Flag bit can therefore be expressed as:

$$INT = I1\bullet CR16 + I2\bullet CR26 + I3\bullet CR36$$

where INT = Composite Interrupt Flag (Bit 7)
I1 = Timer #1 Interrupt Flag (Bit 0)
I2 = Timer #2 Interrupt Flag (Bit 1)
I3 = Timer #3 Interrupt Flag (Bit 2)

An interrupt flag is cleared by a Timer Reset condition, i.e., External $\overline{RESET}=0$ or Internal Reset Bit (CR10)=1. It will also be cleared by a Read Timer Counter Command provided that the Status Register has previously been read while the interrupt flag was set. This condition on the Read Status Register-Read Timer Counter (RS-RT) sequence is designed to prevent missing interrupts which might occur after the status register is read, but prior to reading the Timer Counter.

An Individual Interrupt Flag is also cleared by a Write Timer Latches (W) command or a Counter Initialization (CI) sequence, provided that W or CI affects the Timer corresponding to the individual Interrupt Flag

COUNTER LATCH INITIALIZATION

Each of the three independent timers consists of a 16-bit addressable counter and a 16-bit addressable latch. The counters are preset to the binary numbers stored in the latches Counter initialization results in the transfer of the latch contents to the counter. See notes in Table 4 regarding the binary number N, L, or M placed into the Latches and their relationship to the output waveforms and counter Time-Outs

Since the PTM data bus is 8-bits wide and the counters are 16-bits wide, a temporary register (MSB Buffer Register) is provided This "write only" register is for the Most-Significant Byte of the desired latch data. Three addresses are provided for the MSB Buffer Register (as indicated in Table 1), but they all lead to the same Buffer. Data from the MSB Buffer will automatically be transferred into the Most-Significant Byte of Timer #X when a Write Timer #X Latches Command is performed So it can be seen that the MC6840 has been designed to allow transfer of two bytes of data into the counter latches provided that the MSB is transferred first The storage order must be observed to ensure proper latch operation.

In many applications, the source of the data will be an M6800 Family MPU It should be noted that the 16-bit store operations of the M6800 family microprocessors (STS and STX) transfer data in the order required by the PTM. A Store Index Register Instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transferred directly into a selected counter latch with a single instruction.

A logic zero at the $\overline{RESET}$ input also initializes the counter latches. In this case, all latches will assume a maximum count of $65{,}535_{10}$. It is important to note that an Internal

TABLE 3 – PTM OPERATING MODE SELECTION

CRX3	CRX4	CRX5	
0	0	0	Continuous Operating Mode. Gate ↓ or Write to Latches or Reset Causes Counter Initialization
1	0	0	Frequency Comparison Mode. Interrupt If Gate ↓‗⌈‾↓ is < Counter Time Out
0	1	0	Continuous Operating Mode· Gate ↓ or Reset Causes Counter Initialization
1	1	0	Pulse Width Comparison Mode· Interrupt if Gate ↓‗‗↑ is < Counter Time Out
0	0	1	Single Shot Mode. Gate ↓ or Write to Latches or Reset Causes Counter Initialization
1	0	1	Frequency Comparison Mode· Interrupt If Gate ↓‗⌈‾↓ is > Counter Time Out
0	1	1	Single Shot Mode: Gate ↓ or Reset Causes Counter Initialization
1	1	1	Pulse Width Comparison Mode. Interrupt If Gate ↓‗‗↓ is > Counter Time Out

Reset (Bit zero of Control Register 1 Set) has no effect on the counter latches.

COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with subsequent clearing of the Individual Interrupt Flag associated with the counter Counter Initialization always occurs when a reset condition ($\overline{\text{RESET}}=0$ or CR10 = 1) is recognized It can also occur – depending on Timer Mode – with a Write Timer Latches command or recognition of a negative transition of the Gate input

Counter recycling or re-initialization occurs when a negative transition of the clock input is recognized after the counter has reached an all-zero state In this case, data is transferred from the Latches to the Counter

ASYNCHRONOUS INPUT/OUTPUT LINES

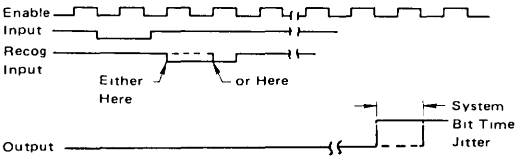
Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line The inputs are high-impedance, TTL-compatible lines and ouputs are capable of driving two standard TTL loads

CLOCK INPUTS ($\overline{\text{C1}}$, $\overline{\text{C2}}$, and $\overline{\text{C3}}$) – Input pins $\overline{\text{C1}}$, $\overline{\text{C2}}$, and $\overline{\text{C3}}$ will accept asynchronous TTL voltage level signals to decrement Timers 1, 2, and 3, respectively. The high and low levels of the external clocks must each be stable for at least one system clock period plus the sum of the setup and hold times for the clock inputs The asynchronous clock rate can vary from dc to the limit imposed by the Enable Clock Setup, and Hold times

The external clock inputs are clocked in by Enable pulses Three Enable periods are used to synchronize and process the external clock The fourth Enable pulse decrements the internal counter This does not affect the input frequency, it merely creates a delay between a clock input transition and internal recognition of that transition by the PTM All references to C inputs in this document relate to internal recognition of the input transition Note that a clock high or low level which does not meet setup and hold time specifications may require an additional Enable pulse for recognition. When observing recurring events, a lack of synchronization will result in "jitter" being observed on the output of the PTM when using asynchronous clocks and gate input signals There are two types of jitter "System jitter" is the result of the input signals being out of synchronization with Enable, permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or the subsequent bit time

"Input jitter" can be as great as the time between input signal negative going transitions plus the system jitter, if the first transition is recognized during one system cycle, and not recognized the next cycle, or vice versa See Figure 9

FIGURE 9 – INPUT JITTER



CLOCK INPUT $\overline{\text{C3}}$ (÷8 PRESCALER MODE) – External clock input $\overline{\text{C3}}$ represents a special case when Timer #3 is programmed to utilize its optional ÷8 prescaler mode

The divide-by-8 prescaler contains an asynchronous ripple counter, thus, input setup (t_{su}) and hold times (t_{hd}) do not apply As long as minimum input pulse widths are maintained, the counter will recognize and process all input clock ($\overline{\text{C3}}$) transitions However, in order to guarantee that a clock transition is processed during the current E cycle, a certain amount of synchronization time (t_{sync}) is required between the $\overline{\text{C3}}$ transition and the falling edge of Enable (see Figure 9) If the synchronization time requirement is not met, it is possible that the $\overline{\text{C3}}$ transition will not be processed until the following E cycle

The maximum input frequency and allowable duty cycles for the ÷8 prescaler mode are specified under the AC Operating Characteristics Internally, the ÷8 prescaler output is treated in the same manner as the previously discussed clock inputs

GATE INPUTS ($\overline{\text{G1}}$, $\overline{\text{G2}}$, $\overline{\text{G3}}$) – Input pins $\overline{\text{G1}}$, $\overline{\text{G2}}$, and $\overline{\text{G3}}$ accept asynchronous TTL-compatible signals which are used as triggers or clock gating functions to Timers 1, 2, and 3, respectively The gating inputs are clocked into the PTM by the E (enable) clock in the same manner as the previously discussed clock inputs That is, a Gate transition is recognized by the PTM on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the Gate input must be stable for at least one system clock period plus the sum of setup and hold times All references to G transition in this document relate to internal recognition of the input transition.

The Gate inputs of all timers directly affect the internal 16-bit counter. The operation of $\overline{\text{G3}}$ is therefore independent of the ÷8 prescaler selection

TIMER OUTPUTS (O1, O2, O3) – Timer outputs O1, O2, and O3 are capable of driving up to two TTL loads and produce a defined output waveform for either Continuous or Single-Shot Timer modes Output waveform definition is accomplished by selecting either Single 16-bit or Dual 8-bit operating modes The Single 16-bit mode will produce a square-wave output in the continuous mode and a single pulse in the single-shot mode The Dual 8-bit mode will produce a variable duty cycle pulse in both the continuous and single-shot timer modes. One bit of each Control Register (CRX7) is used to enable the corresponding output. If this bit is cleared, the output will remain low (V_{OL}) regardless of the operating mode. If it is cleared while the output is high the output will go low during the first enable cycle following a write to the Control Register

The Continuous and Single-Shot Timer Modes are the only ones for which output response is defined in this data sheet Refer to the Programmable Timer Fundamentals and Applications manual for a discussion of the output signals in other modes Signals appear at the outputs (unless CRX7 = 0) during Frequency and Pulse Width comparison modes, but the actual waveform is not predictable in typical applications.

TIMER OPERATING MODES

The MC6840 has been designed to operate effectively in a wide variety of applications. This is accomplished by using three bits of each control register (CRX3, CRX4, and CRX5) to define different operating modes of the Timers These modes are divided into WAVE SYNTHESIS and WAVE MEASUREMENT modes, and are outlined in Table 4

TABLE 4 – OPERATING MODES

Control Register			Timer Operating Mode	
CRX3	CRX4	CRX5		
0	*	0	Continuous	Synthesizer
0	*	1	Single-Shot	
1	0	*	Frequency Comparison	Measurement
1	1	*	Pulse Width Comparison	

*Defines Additional Timer Function Selection

One of the WAVE SYNTHESIS modes is the Continuous Operating mode, which is useful for cyclic wave generation Either symmetrical or variable duty-cycle waves can be generated in this mode. The other wave synthesis mode, the Single-Shot mode, is similar in use to the Continuous operating mode, however, a single pulse is generated, with a programmable preset width

The WAVE MEASUREMENT modes include the Frequency Comparison and Pulse Width Comparison modes which are used to measure cyclic and singular pulse widths, respectively

In addition to the four timer modes in Table 4, the remaining control register bit is used to modify counter initialization and enabling or interrupt conditions

WAVE SYNTHESIS MODES

CONTINUOUS OPERATING MODE (TABLE 5) – The continuous mode will synthesize a continuous wave with a period proportional to the preset number in the particular timer latches Any of the timers in the PTM may be programmed to operate in a continuous mode by writing zeroes into bits 3 and 5 of the corresponding control register Assuming that the timer output is enabled (CRX7 = 1), either a square wave or a variable duty cycle waveform will be generated at the Timer Output, OX The type of output is selected via Control Register Bit 2

Either a Timer Reset (CR10 = 1 or External Reset = 0) condition or internal recognition of a negative transition of the Gate input results in Counter Initialization A Write Timer latches command can be selected as a Counter Initialization signal by clearing CRX4

The counter is enabled by an absence of a Timer Reset condition and a logic zero at the Gate input In the 16-bit mode, the counter will decrement on the first clock cycle during or after the counter initialization cycle It continues to decrement on each clock signal so long as G remains low and no reset condition exists A Counter Time Out (the first clock after all counter bits = 0) results in the Individual Interrupt Flag being set and reinitialization of the counter

In the Dual 8-bit mode (CRX2 = 1) [refer to the example in Figure 10 and Tables 5 and 6] the MSB decrements once for every full countdown of the LSB + 1 When the LSB = 0, the MSB is unchanged, on the next clock pulse the LSB is reset to the count in the LSB Latches, and the MSB is decremented by 1 (one) The output, if enabled, remains low during and after initialization and will remain low until the counter MSB is all zeroes The output will go high at the beginning of the next clock pulse The output remains high until both the LSB and MSB of the counter are all zeroes At the beginning of the next clock pulse the defined Time Out (TO) will occur and the output will go low In the Dual 8-bit mode the period of the output of the example in **Figure 10** would span 20 clock pulses as opposed to 1546 clock pulses using the normal 16-bit mode

A special time-out condition exists for the dual 8-bit mode (CRX2 = 1) if L = 0 In this case, the counter will revert to a mode similar to the single 16-bit mode, except Time Out occurs after M + 1* clock pulses The output, if enabled, goes low during the Counter Initialization cycle and reverses state at each Time Out The counter remains cyclical (is reinitialized at each Time Out) and the Individual Interrupt Flag is set when Time Out occurs If M = L = 0, the internal counters do not change, but the output toggles at a rate of ½ the clock frequency

TABLE 5 – CONTINUOUS OPERATING MODES

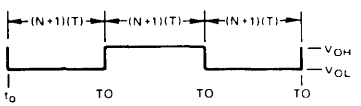
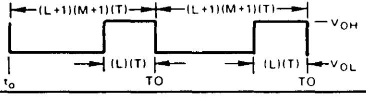
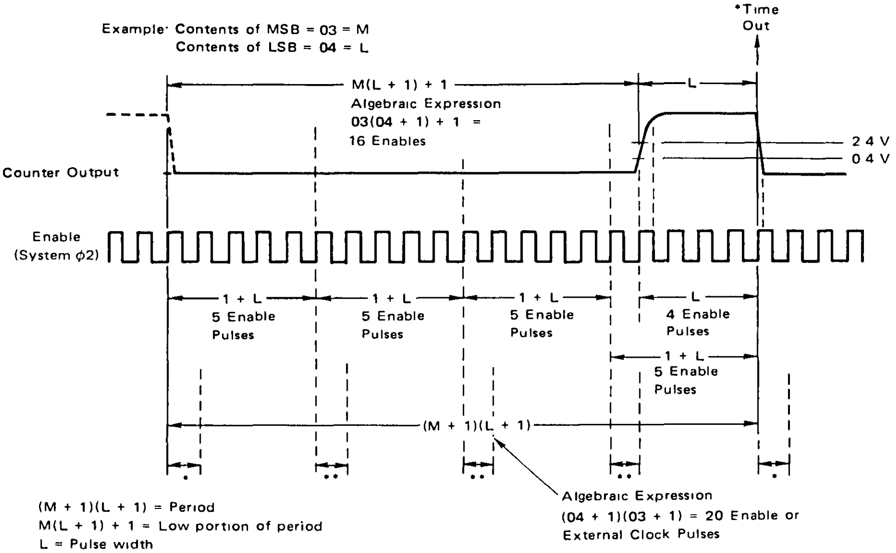
Synthesis Modes		CONTINUOUS MODE (CRX3 = 0, CRX5 = 0)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = 1)
0	0	$\overline{G}\downarrow$+W+R	 (N+1)(T) (N+1)(T) (N+1)(T) V_{OH} V_{OL} t_0 TO TO TO
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	 (L+1)(M+1)(T) (L+1)(M+1)(T) V_{OH} (L)(T) (L)(T) V_{OL} t_0 TO TO
1	1	$\overline{G}\downarrow$+R	

FIGURE 10 — TIMER OUTPUT WAVEFORM EXAMPLE
(Continuous Dual 8-Bit Mode Using Internal Enable)


Example: Contents of MSB = 03 = M
Contents of LSB = 04 = L
*Time Out
M(L + 1) + 1
Algebraic Expression
03(04 + 1) + 1 =
16 Enables
L
2 4 V
0 4 V
Counter Output
Enable
(System φ2)
1 + L
5 Enable Pulses
1 + L
5 Enable Pulses
1 + L
5 Enable Pulses
L
4 Enable Pulses
1 + L
5 Enable Pulses
(M + 1)(L + 1)
*
**
**
**
*
Algebraic Expression
(04 + 1)(03 + 1) = 20 Enable or External Clock Pulses
(M + 1)(L + 1) = Period
M(L + 1) + 1 = Low portion of period
L = Pulse width

*Preset LSB and MSB to Respective Latches on the negative transition of the Enable

**Preset LSB to LSB Latches and Decrement MSB by one on the negative transition of the Enable

The discussion of the Continuous Mode has assumed that the application requires an output signal It should be noted that the Timer operates in the same manner with the output disabled (CRX7 = 0) A Read Timer Counter command is valid regardless of the state of CRX7

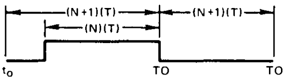
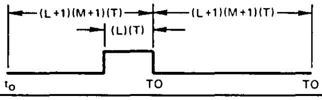
SINGLE-SHOT TIMER MODE — This mode is identical to the Continuous Mode with three exceptions The first of these is obvious from the name — the output returns to a low level after the initial Time Out and remains low until another Counter Initialization cycle occurs

As indicated in Table 6, the internal counting mechanism remains cyclical in the Single-Shot Mode Each Time Out of the counter results in the setting of an Individual Interrupt Flag and re-initialization of the counter

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the Gate input level remaining in the low state for the Single-Shot mode

Another special condition is introduced in the Single-Shot mode If L = M = 0 (Dual 8-bit) or N = 0 (Single 16-bit), the output goes low on the first clock received during or after Counter Initialization The output remains low until the Operating Mode is changed or nonzero data is written into the Counter Latches Time Outs continue to occur at the end of each clock period

TABLE 6 — SINGLE-SHOT OPERATING MODES

Synthesis Modes		SINGLE-SHOT MODE (CRX3 = 0, CRX7 = 1, CRX5 = 1)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	Timer Output (OX)
0	0	$\overline{G}\downarrow$+W+R	 (N+1)(T) (N+1)(T) (N)(T) t0 TO TO
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	 (L+1)(M+1)(T) (L+1)(M+1)(T) (L)(T) t0 TO TO
1	1	$\overline{G}\downarrow$+R	

Symbols are as defined in Table 5.

The three differences between Single-Shot and Continous Timer Mode can be summarized as attributes of the Single-Shot mode

1 Output is enabled for only one pulse until it is reinitialized

2. Counter Enable is independent of Gate

3 L=M=0 or N=0 disables output

Aside from these differences, the two modes are identical

WAVE MEASUREMENT MODES

TIME INTERVAL MODES – The Time Interval Modes are the Frequency (period) Measurement and Pulse Width Comparison Modes, and are provided for those applications which require more flexibility of interrupt generation and Counter Initialization. Individual Interrupt Flags are set in these modes as a function of both Counter Time Out and transitions of the Gate input Counter Initialization is also affected by Interrupt Flag status

A timer's output is normally not used in a Wave Measurement mode, but it is defined If the output is enabled, it will operate as follows During the period between reinitialization of the timer and the first Time Out, the output will be a logical zero If the first Time Out is completed (regardless of its method of generation), the output will go high If further TO's occur, the output will change state at each completion of a Time-Out

The counter does operate in either Single 16-bit or Dual 8-bit modes as programmed by CRX2 Other features of the Wave Measurement Modes are outlined in Table 7

Frequency Comparison Or Period Measurement Mode (CRX3=1, CRX4=0) – The Frequency Comparison Mode with CRX5=1 is straightforward If Time Out occurs prior to the first negative transition of the Gate input after a Counter Initialization cycle, an Individual Interrupt Flag is set The counter is disabled, and a Counter Initialization cycle cannot begin until the interrupt flag is cleared and a negative transition on $\overline{G}$ is detected.

If CRX5=0, as shown in Tables 7 and 8, an interrupt is generated if Gate input returns low prior to a Time Out If a Counter Time Out occurs first, the counter is recycled and continues to decrement A bit is set within the timer on the initial Time Out which precludes further individual interrupt generation until a new Counter Initialization cycle has been completed. When this internal bit is set, a negative transition of the Gate input starts a new Counter Initialization cycle (The condition of $\overline{G}\downarrow \cdot \overline{I} \cdot TO$ is satisfied, since a Time Out has occurred and no individual Interrupt has been generated)

Any of the timers within the PTM may be programmed to compare the period of a pulse (giving the frequency after calculations) at the Gate input with the time period requested for Counter Time Out A negative transition of the Gate Input enables the counter and starts a Counter Initialization cycle – provided that other conditions, as noted in Table 8, are satisfied The counter decrements on each clock signal recognized during or after Counter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs It can be seen from Table 8 that an interrupt condition will be generated if CRX5=0 and the period of the pulse (single pulse or measured separately repetitive pulses) at the Gate input is less than the Counter Time Out period If CRX5=1, an interrupt is generated if the reverse is true

Assume now with CRX5=1 that a Counter Initialization has occurred and that the Gate input has returned low prior to Counter Time Out Since there is no Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle The process will continue with frequency comparison being performed on each Gate input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit

Pulse Width Comparison Mode (CRX3=1, CRX4=1) – This mode is similar to the Frequency Comparison Mode except for a positive, rather than negative, transition of the Gate input terminates the count With CRX5=0, an Individual Interrupt Flag will be generated if the zero level pulse applied to the Gate input is less than the time period required for Counter Time Out With CRX5=1, the interrupt is generated when the reverse condition is true

As can be seen in Table 8, a positive transition of the Gate input disables the counter With CRX5=0, it is therefore possible to directly obtain the width of any pulse causing an interrupt Similar data for other Time Interval Modes and conditions can be obtained, if two sections of the PTM are dedicated to the purpose

FIGURE 7 – OUTPUT DELAY

CRX3 = 1			
CRX4	**CRX5**	**Application**	**Condition for Setting Individual Interrupt Flag**
0	0	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is greater than Counter Time Out (TO)

TABLE 8 — FREQUENCY COMPARISON MODE

Mode	Bit 3	Bit 4	Control Reg. Bit 5	Counter Initialization	Counter Enable Flip-Flop Set (CE)	Counter Enable Flip-Flop Reset (CE)	Interrupt Flag Set (I)
Frequency Comparison	1	0	0	$\overline{G}\downarrow\cdot\overline{I}\cdot(\overline{CE}+TO)+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	W + R + I	$\overline{G}\downarrow$ Before TO
	1	0	1	$\overline{G}\downarrow\cdot\overline{I}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	W + R + I	TO Before $\overline{G}\downarrow$
Pulse Width Comparison	1	1	0	$\overline{G}\downarrow\cdot\overline{I}+\overline{R}$	$\overline{G}\downarrow\overline{W}\cdot\overline{R}\cdot\overline{I}$	W + R + I + G	$\overline{G}\downarrow$ Before TO
	1	1	1	$\overline{G}\downarrow\cdot\overline{I}+\overline{R}$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	W + R + I + G	TO Before $\overline{G}\downarrow$

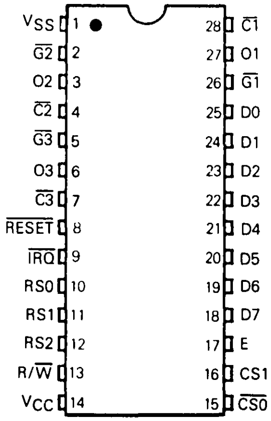
$\overline{G}\downarrow$ = Negative transition of Gate input
W = Write Timer Latches Command
R = Timer Reset (CR10 = 1 or External $\overline{RESET}$ = 0)
N = 16-Bit Number in Counter Latch
TO = Counter Time Out (All Zero Condition)
I = Interrupt for a given timer

*All time intervals shown above assume the Gate ($\overline{G}$) and Clock ($\overline{C}$) signals are sycnhronized to the system clock (E) with the specified setup and hold time requirements

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6840P
	1.0 MHz	−40°C to +85°C	MC6840CP
	1.5 MHz	0°C to 70°C	MC68A40P
	1.5 MHz	−40°C to +85°C	MC68A40CP
	2 0 MHz	0°C to 70°C	MC68B40P
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6840S
	1.0 MHz	−40°C to +85°C	MC6840CS
	1.5 MHz	0°C to 70°C	MC68A40S
	1.5 MHz	−40°C to +85°C	MC68A40CS
	2.0 MHz	0°C to 70°C	MC68B40S

PIN ASSIGNMENT


VSS 1
$\overline{G2}$ 2
O2 3
$\overline{C2}$ 4
$\overline{G3}$ 5
O3 6
$\overline{C3}$ 7
$\overline{RESET}$ 8
$\overline{IRQ}$ 9
RS0 10
RS1 11
RS2 12
R/$\overline{W}$ 13
VCC 14
28 $\overline{C1}$
27 O1
26 $\overline{G1}$
25 D0
24 D1
23 D2
22 D3
21 D4
20 D5
19 D6
18 D7
17 E
16 CS1
15 $\overline{CS0}$

MC6844

Direct Memory Access Controller (DMAC)

The MC6844 Direct Memory Access Controller (DMAC) performs the function of transferring data directly between memory and peripheral device controllers. It directly transfers the data by controlling the address and data bus in place of an MPU in a bus organized system.

The bus interface of the MC6844 includes select, read/write, interrupt, transfer request/grant, a data port, and an address port which allow data transfer over an 8-bit bidirectional data bus. The functional configuration of the DMAC is programmed via the data bus. The internal structure provides for control and handling of four individual channels, each of which is separately configured. Programmable control registers provide control for data transfer location and data block length, individual channel control and transfer mode configuration, priority of channel servicing, data chaining, and interrupt control. Status and control lines provide control to peripheral controllers.

The mode of transfer for each channel can be programmed as one of two single-byte transfer modes or a burst transfer mode.

Typical MC6844 applications are a Floppy Disk Controller (FDC) and an Advanced Data Link Controller (ADLC) DMA interface.

MC6844 features include:

- Four DMA Channels, Each Having a 16-Bit Address Register and a 16-Bit Byte Count Register
- 2 M Byte/Sec Maximum Data Transfer Rate
- Selection of Fixed or Rotating Priority Service Control
- Separate Control Bits for Each Channel
- Data Chain Function
- Address Increment or Decrement Update
- Programmable Interrupts and DMA End to Peripheral Controllers

FIGURE 1 – M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM

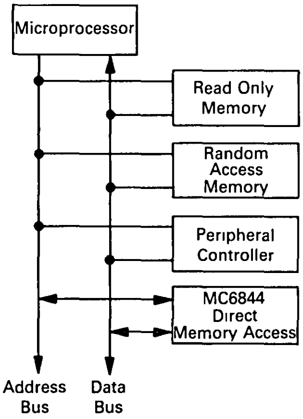
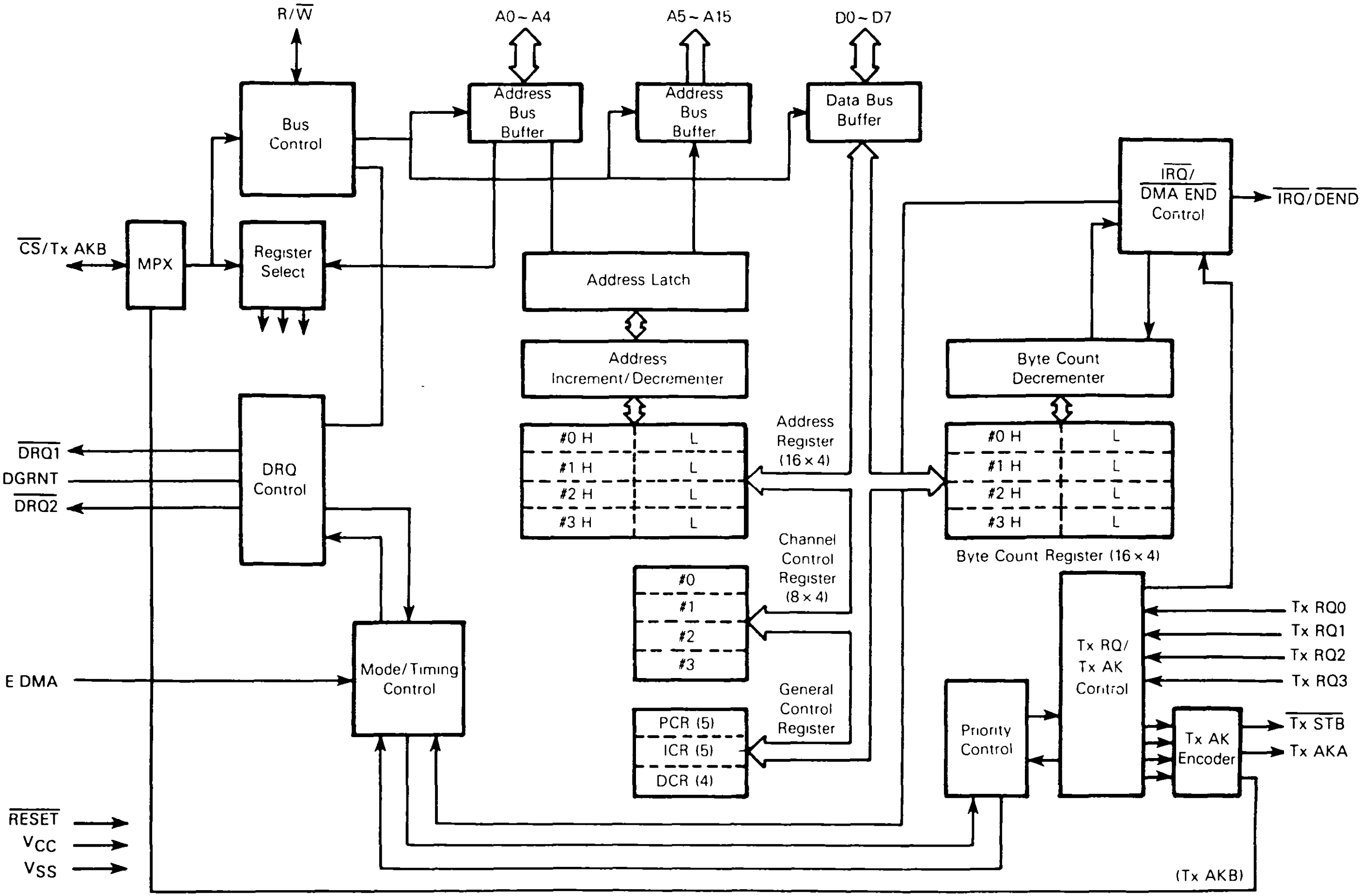


FIGURE 2 — BLOCK DIAGRAM OF DMAC


R/$\overline{W}$
A0 ~ A4
A5 ~ A15
D0 ~ D7
Bus Control
Address Bus Buffer
Address Bus Buffer
Data Bus Buffer
$\overline{IRQ}$/ $\overline{DMA\ END}$ Control
$\overline{IRQ}$/$\overline{DEND}$
$\overline{CS}$/Tx AKB
MPX
Register Select
Address Latch
Address Increment/Decrementer
Byte Count Decrementer
#0 H
#1 H
#2 H
#3 H
L
Address Register (16 × 4)
Byte Count Register (16 × 4)
$\overline{DRQ1}$
DGRNT
$\overline{DRQ2}$
DRQ Control
Channel Control Register (8 × 4)
#0
#1
#2
#3
General Control Register
PCR (5)
ICR (5)
DCR (4)
E DMA
Mode/Timing Control
Priority Control
Tx RQ/ Tx AK Control
Tx RQ0
Tx RQ1
Tx RQ2
Tx RQ3
Tx AK Encoder
$\overline{Tx\ STB}$
Tx AKA
$\overline{RESET}$
V_{CC}
V_{SS}
(Tx AKB)

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage	V_{in}*	−0.3 to +7.0	V
Operating Temperature Range MC6844, MC68A44, MC68B44 MC6844C, MC68A44C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 100 60	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{PORT}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

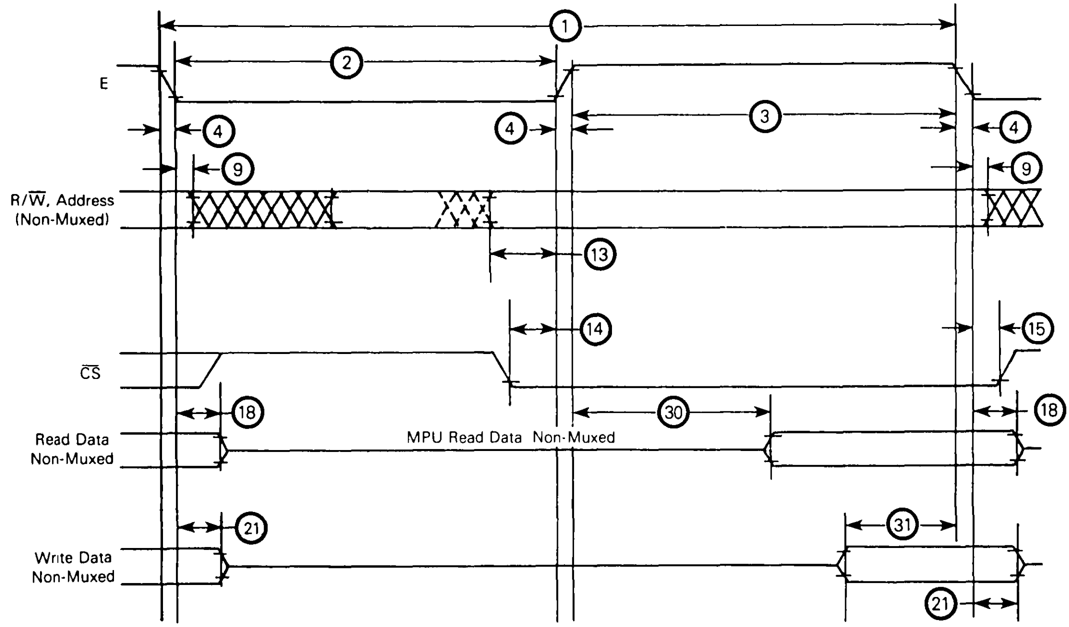
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	V_{SS}+2.0	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$/Tx AKB Other Inputs	V_{IL}	V_{SS}−0.3 V_{SS}−0.3	– –	V_{SS}+0.6 V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	Tx RQ0-3, E, $\overline{RESET}$, DGRNT	I_{in}	–	–	2.5	µA
Hi-Z Leakage Current (V_{in} = 0.4 to 2.4 V)	A0-A15, R/$\overline{W}$ D0-D7	I_{TSI}	−10	–	10	µA
Output High Voltage (I_{Load} = −205 µA (I_{Load} = −145 µA) (I_{Load} = −100 µA)	 D0-D7 A0-A15, R/$\overline{W}$ All Others	V_{OH}	 V_{SS}+2.4 V_{SS}+2.4 V_{SS}+2.4	 – – –	 – – –	V
Output Low Voltage (I_{Load} = 1.6 mA)	All Others	V_{OL}	–	–	V_{SS}+0.4	V
Source Current (V_{in} = 0 V, Figure 10)	$\overline{CS}$/Tx AKB	I_{CSS}	–	10	16	mA
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	500	750*	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	E D0-D7, CS, A0-A4, R/$\overline{W}$ All Others	C_{in}	– – –	– – –	20 12.5 10	pF
		C_{out}	–	–	12	pF

* For temperatures less than T_A = 0°C, P_{INT} maximum will increase

MPU MODE TIMING (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6844		MC68A44		MC68B44		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	TBD		ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	–	20	–	20	–	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	TBD		ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 3 — MPU MODE TIMING


①
②
E
④
④
③
④
⑨
⑨
R/W̄, Address (Non-Muxed)
⑬
⑭
⑮
C̄S̄
⑱
㉚
⑱
Read Data Non-Muxed
MPU Read Data Non-Muxed
㉑
㉛
Write Data Non-Muxed
㉑

NOTES
1 Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FIGURE 4 — MODE 1 TIMING
(TSC STEAL MODE)

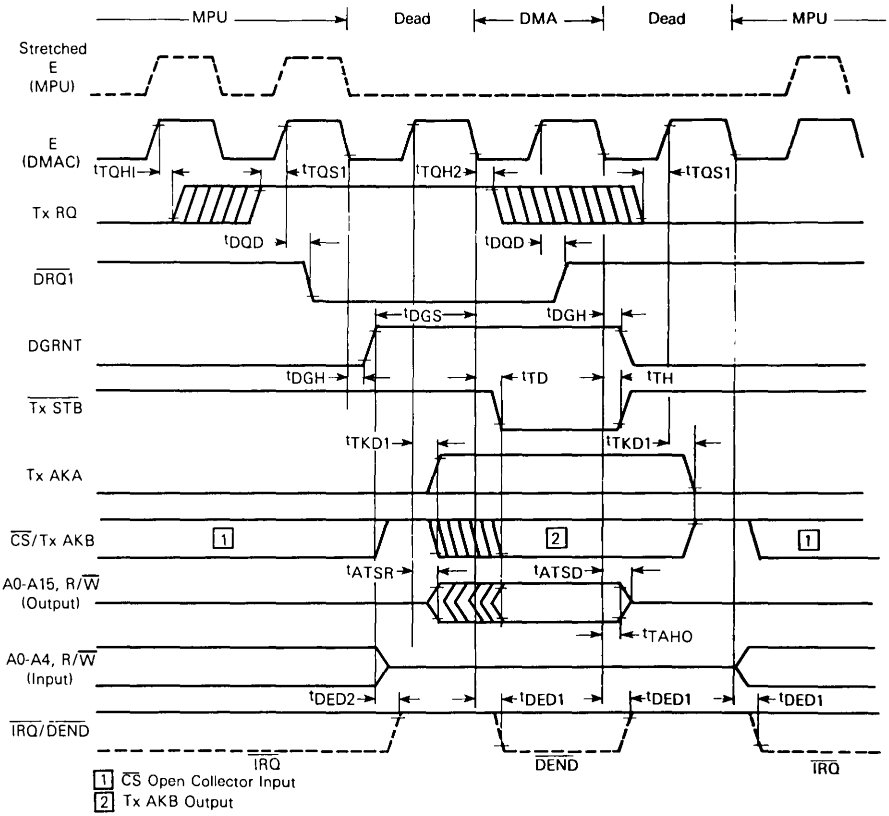
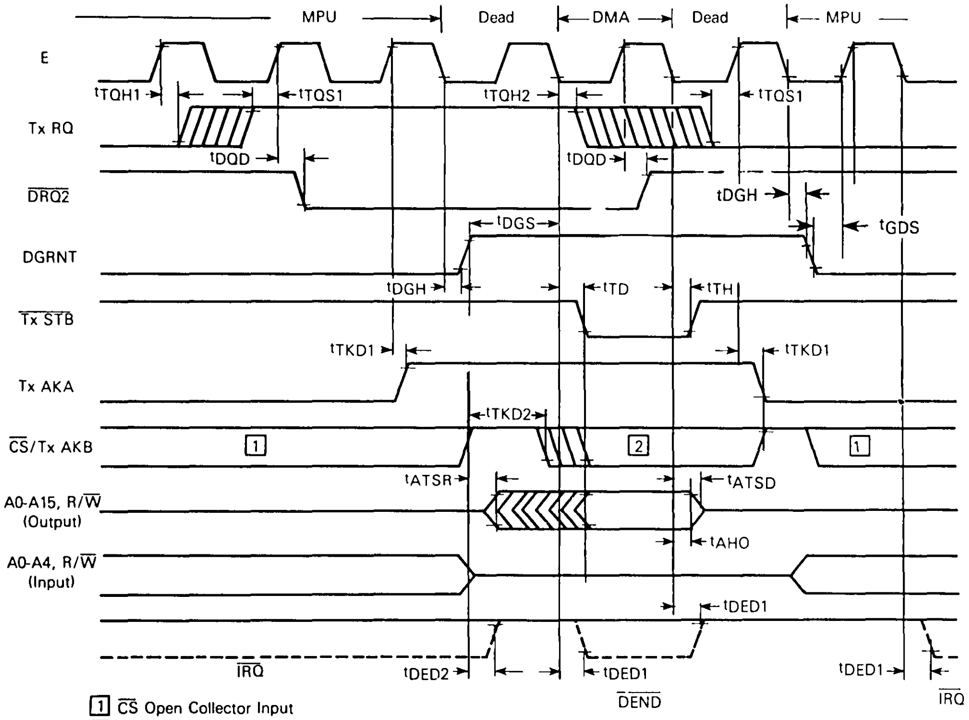
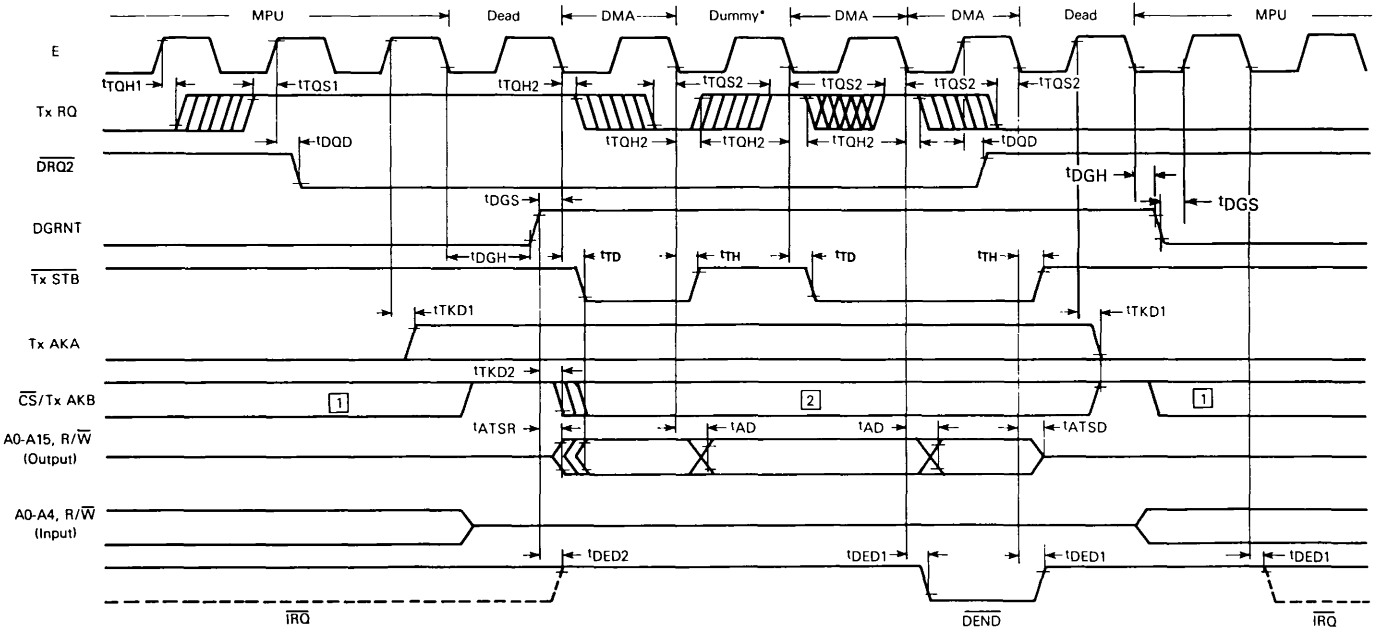

MPU
Dead
DMA
Dead
MPU
Stretched E (MPU)
E (DMAC)
Tx RQ
DRQ1
DGRNT
Tx STB
Tx AKA
CS/Tx AKB
A0-A15, R/W (Output)
A0-A4, R/W (Input)
IRQ/DEND
tTQHI
tTQS1
tTQH2
tTQS1
tDQD
tDQD
tDGS
tDGH
tDGH
tTD
tTH
tTKD1
tTKD1
tATSR
tATSD
tTAHO
tDED2
tDED1
tDED1
tDED1
1
2
1
IRQ
DEND
IRQ
1 CS Open Collector Input
2 Tx AKB Output

FIGURE 5 — MODE 2 TIMING
(HALT STEAL MODE)


MPU
Dead
DMA
Dead
MPU
E
tTQH1
tTQS1
tTQH2
tTQS1
Tx RQ
tDQD
tDQD
$\overline{\text{DRQ2}}$
tDGH
tDGS
tGDS
DGRNT
tDGH
tTD
tTH
$\overline{\text{Tx STB}}$
tTKD1
tTKD1
Tx AKA
tTKD2
$\overline{\text{CS}}$/Tx AKB
1
2
1
tATSR
tATSD
A0-A15, R/$\overline{\text{W}}$
(Output)
tAHO
A0-A4, R/$\overline{\text{W}}$
(Input)
tDED1
$\overline{\text{IRQ}}$
tDED2
tDED1
tDED1
$\overline{\text{DEND}}$
$\overline{\text{IRQ}}$

1 $\overline{\text{CS}}$ Open Collector Input
2 Tx AKB Output

FIGURE 6 — MODE 3 TIMING
(HALT BURST MODE)


MPU
Dead
DMA
Dummy*
DMA
DMA
Dead
MPU
E
Tx RQ
$\overline{\text{DRQ2}}$
DGRNT
$\overline{\text{Tx STB}}$
Tx AKA
$\overline{\text{CS}}$/Tx AKB
A0-A15, R/$\overline{\text{W}}$ (Output)
A0-A4, R/$\overline{\text{W}}$ (Input)
$\overline{\text{IRQ}}$
$\overline{\text{DEND}}$
$\overline{\text{IRQ}}$
t_{TQH1}
t_{TQS1}
t_{TQH2}
t_{TQS2}
t_{TQS2}
t_{TQS2}
t_{TQS2}
t_{DQD}
t_{TQH2}
t_{TQH2}
t_{TQH2}
t_{DQD}
t_{DGH}
t_{DGS}
t_{DGS}
t_{DGH}
t_{TD}
t_{TH}
t_{TD}
t_{TH}
t_{TKD1}
t_{TKD1}
t_{TKD2}
1
2
1
t_{ATSR}
t_{AD}
t_{AD}
t_{ATSD}
t_{DED2}
t_{DED1}
t_{DED1}
t_{DED1}

1 $\overline{\text{CS}}$ Open Collector Input

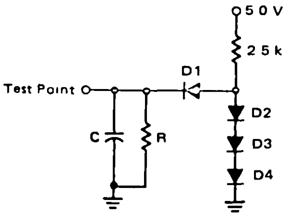
2 Tx AKB Output

*No transfer (dummy cycle) because Tx RQ was negated at start of E cycle

DMA TIMING (Load Condition Figure 7)

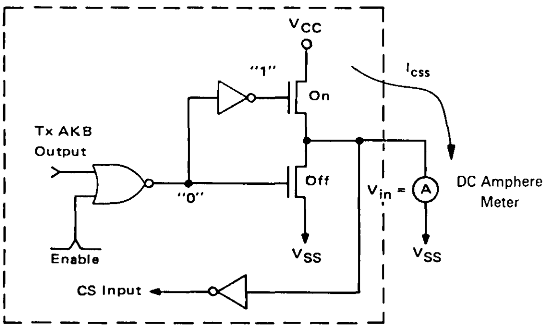
Characteristic		Symbol	MC6844		MC68A44		MC68B44		Unit
			Min	Max	Min	Max	Min	Max	
Tx RQ Setup Time									
E Rising Edge		t_{TQS1}	120	–	120	–	120	–	ns
E Falling Edge		t_{TQS2}	210	–	210	–	170	–	
Tx RQ Hold Time									
E Rising Edge		t_{TQH1}	20	–	10	–	10	–	ns
E Falling Edge		t_{TQH2}	20	–	10	–	10	–	
DGRNT Setup Time		t_{DGS}	155	–	125	–	115	–	ns
DGRNT Hold Time		t_{DGH}	10	–	10	–	10	–	ns
Address Output Delay Time	A0-A15, R/$\overline{W}$	t_{AD}	–	270	–	180	–	150	ns
Address Output Hold Time	A0-A15, R/$\overline{W}$	t_{AHO}	30	–	20	–	20	–	ns
Address Three-State Delay Time	A0-A15, R/$\overline{W}$	t_{ATSD}	–	720	–	460	–	370	ns
Address Three-State Recovery Time		t_{ATSR}	–	430	–	280	–	210	ns
Delay Time	$\overline{DRQ1}$, $\overline{DRQ2}$	t_{DQD}	–	375	–	250	–	200	ns
Tx AK Delay Time									
E Rising Edge		t_{TKD1}	–	400	–	310	–	250	ns
DGRNT Rising Edge		t_{TKD2}	–	190	–	160	–	145	
$\overline{IRQ}$/$\overline{DEND}$ Delay Time									
E Falling Edge		t_{DED1}	–	300	–	250	–	230	ns
DGRNT Rising Edge		t_{DED2}	–	190	–	160	–	145	
Tx $\overline{STB}$ Output Delay Time		t_{TD}	–	270	–	180	–	150	ns
Tx STB Output Hold Time		t_{TH}	30	–	20	–	20	–	ns

FIGURE 7 — TEST LOADS



Test Pin	C = pF	R = kΩ
D0-D7	130	11.7
A0-A15, R/$\overline{W}$	90	16.5
$\overline{CS}$/Tx AKB	50	24
Others	30	24

FIGURE 8 — CS/Tx AKB SOURCE CURRENT TEST CIRCUIT



INTRODUCTION

The MC6844 DMAC has four DMA channels which can be independently configured by software using fifteen addressable registers. Eight of the addressable registers are 16-bit registers, and seven are 8-bit registers. Associated with each channel are a 16-bit Address Register, a 16-bit Byte Control Register, and an 8-bit Channel Control Register.The DMAC also has three 8-bit registers which affect all of the channels: the Priority Control Register, the Interrupt Control Register, and the Data Chain Register. A block diagram of the DMAC is presented in Figure 2.

SOFTWARE INITIALIZATION

A channel is initialized for DMA by loading the channel address register with the desired starting DMA address and the channel byte control register with the number of bytes to be transferred. In addition, the channel control register must be initialized for the direction of data transfer, for address register increment or decrement after each byte transfer, and for DMA transfer mode

Each channel can be initialized for one of three transfer modes: Mode 1, Mode 2, or Mode 3. Two read-only status bits in the channel control register indicate when the channel is busy transferring a block of data and when the DMA transfer of a block of data is complete.

The priority control register, the interrupt control register, and the data chain registers must also be initialized.

The priority control register enables/disables each channel and determines whether channel service requests are serviced in a fixed or a rotating priority. The interrupt control register controls assertion of $\overline{\mathrm{IRQ}}$ interrupt by each channel at the end of a data block transfer and sets a flag when $\overline{\mathrm{IRQ}}$ is asserted. The data chain register controls selection of two or four channel operation, selection of data chaining operation, and the channel to be updated in the data chaining mode

When data chaining is enabled, the contents of the channel 3 address and byte count registers are stored into the corresponding registers of the channel selected for chaining after the channel data block transfer is completed.This feature allows for repetitively reading or writing a block of memory

HARDWARE INITIALIZATION

At power-on reset (POR) and anytime $\overline{\mathrm{RESET}}$ is asserted, all device registers except the address and byte count registers are cleared. Therefore, the state of the DMAC after reset is as follows·

- all DMA channels are disabled,
- all interrupts are disabled,
- all flags are cleared,
- address register increment is selected for each channel,
- mode 2 is selected for each channel,
- peripheral controller write-to-memory is selected for each channel,
- two-channel operation is selected, and
- data chaining is disabled.

DMAC BUS CONTROL

During DMA operation, the DMAC controls the system address and data buses and generates system R/$\overline{\mathrm{W}}$. The DMAC also generates $\overline{\mathrm{Tx\ STB}}$, which can be used to derive system VMA; Tx AKA and Tx AKB, which can be used to identify which DMA channel is in service; $\overline{\mathrm{DRQ1}}$ and $\overline{\mathrm{DRQ2}}$, which are used for handshaking with the system MPU, $\overline{\mathrm{DEND}}$, which is asserted when the last byte of a data block is being transferred, and $\overline{\mathrm{IRQ}}$, which when enabled will interrupt the system MPU when a data block transfer is completed. Data itself does not pass through the DMAC, but is transferred between memory and peripheral under control of the DMAC.

TRANSFER MODES

Each DMAC channel can be programmed to operate in one of three modes.* Two of the modes, mode 1 and mode 2, are single-byte transfer modes in which the DMAC returns the bus to the MPU after each DMA transfer by negating the appropriate DMA Request ($\overline{\mathrm{DRQ1}}$ or $\overline{\mathrm{DRQ2}}$). These modes are intended to be used in applications requiring the MPU to regain control of the bus after each byte transfer. Timing information for modes 1 and 2 is presented in Figures 4 and 5.

Mode 3 is a block transfer mode in which the DMAC retains control of the bus until the last byte of the DMA data block has been transferred (byte control register 0), if DGRNT remains asserted during the entire block transfer. In mode 3, byte transfers are possible at the DMAC clock frequency by asserting Tx RQ each cycle. This mode offers the highest DMA transfer rate Mode 3 timing is presented in Figure 6

A flowchart of DMAC operation in each mode is presented in Figure 9.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to the DMAC. The power supply should provide +5 V ±5% to V_{CC}. V_{SS} should be tied to ground. Total power dissipation will not exceed P_D milliwatts.

$\overline{\mathrm{RESET}}$

This input is used to place the DMAC into a known state and provide for an orderly startup procedure. Assertion of $\overline{\mathrm{RESET}}$ clears all internal registers except the address and the byte count registers (see Hardware Initialization).

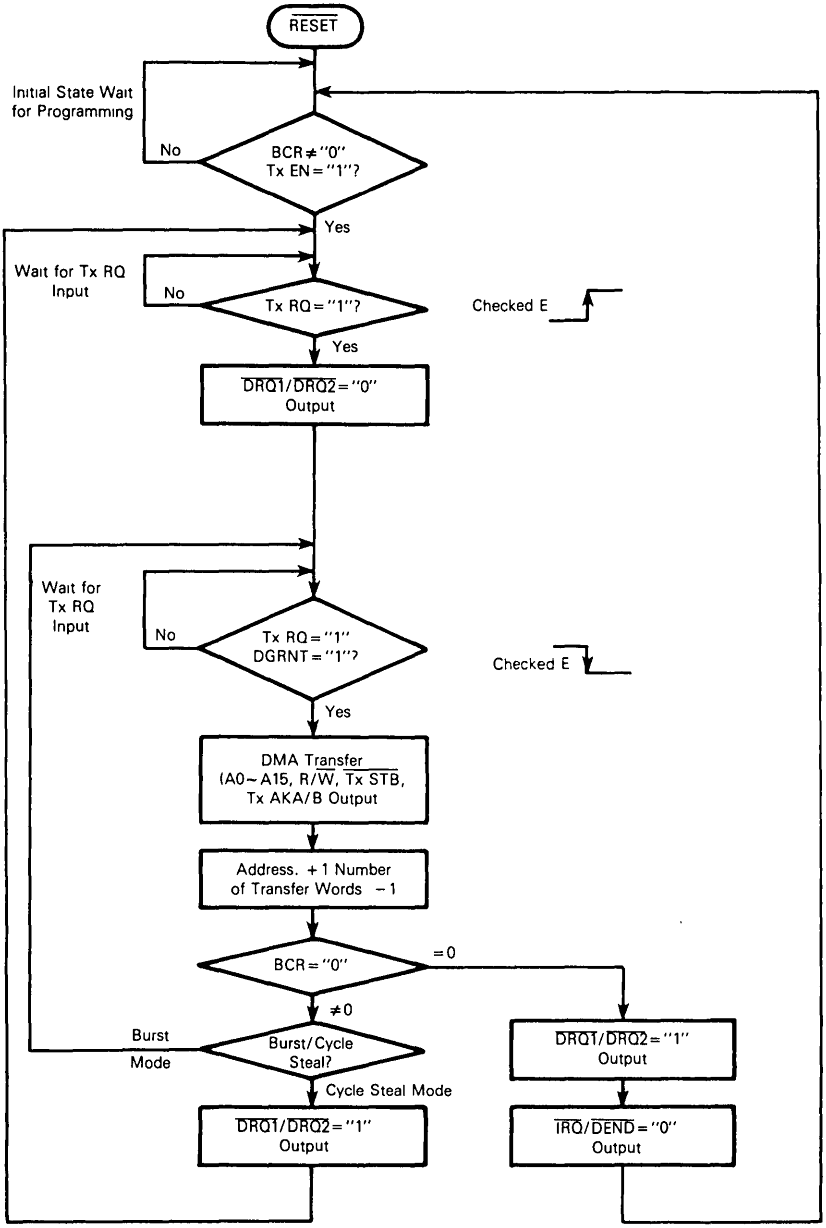
E (ENABLE)

This TTL-compatible input is used to clock the DMAC with the MPU E clock. In systems that perform single-byte transfers by stretching the MPU clock rather than by halting the MPU, the system must be designed to provide a non-stretched E clock to this pin. Clock modules such as the MC6875 are available which provide a separate stretchable E clock to externally-driven MPUs and a non-stretched clock to the DMAC.

---

*Modes 1, 2, and 3 are also called TSC Steal, HALT Steal, and HALT Burst modes

FIGURE 9 — FLOWCHART OF DMAC OPERATION


RESET
Initial State Wait for Programming
No
BCR ≠ "0"
Tx EN = "1"?
Yes
Wait for Tx RQ Input
No
Tx RQ = "1"?
Checked E
Yes
DRQ1/DRQ2 = "0" Output
Wait for Tx RQ Input
No
Tx RQ = "1"
DGRNT = "1"?
Checked E
Yes
DMA Transfer (A0~A15, R/W, Tx STB, Tx AKA/B Output
Address. + 1 Number of Transfer Words − 1
BCR = "0"
= 0
≠ 0
Burst Mode
Burst/Cycle Steal?
Cycle Steal Mode
DRQ1/DRQ2 = "1" Output
DRQ1/DRQ2 = "1" Output
IRQ/DEND = "0" Output

READ/WRITE (R/$\overline{W}$)

This TTL-compatible bidirectional line is a high-impedance input when the DMAC is off the system bus (MPU mode), and an output when the DMAC is controlling the bus (DMA mode). In the MPU mode, this input is used to control the direction of data transfer through the DMAC data bus interface to allow MPU reads and writes to internal registers. In the DMA mode, Read/Write is an output to the system bus, with its state controlled by bit 0 of the appropriate channel control register.

ADDRESS A0-A15

Address lines A0-A4 are bidirectional. In the MPU mode, these lines are inputs used by the MPU to address DMAC registers In the DMA mode, these lines and lines A5-A15 are outputs which assert the contents of the address register of the channel being serviced. Address lines A0-A15 are TTL compatible

DATA D0-D7

These bidirectional TTL-compatible lines are used for data transfer between the MPU and the DMAC. These lines remain in the high-impedance state except when the MPU reads DMAC registers.

INTERRUPT REQUEST/DMA END ($\overline{IRQ}/\overline{DEND}$)

Interrupt Request/DMA End is a TTL-compatible, time-multiplexed, active low output used to interrupt the MPU and signal a peripheral controller when a DMAC data block transfer has ended. $\overline{DEND}$ is asserted during the transfer of the last data byte of a block transfer for one E clock cycle (see Figures 4, 5, and 6). $\overline{IRQ}$ is asserted after the last byte transfer of a block transfer if enabled by setting the proper DEND IRQ enable bit in the interrupt control register (see Table 2). Once asserted, $\overline{IRQ}$ is negated by reading the channel control register of the channel asserting the interrupt.

TRANSFER REQUEST (Tx RQ0-3)

Associated with each channel is a high-impedance input pin used by a peripheral controller to request DMA service by the channel. The Tx RQ pins are sampled by the DMAC in an order of priority determined by the software-programmable state of the priority control register. The Tx RQ pins for channels programmed for mode 1 or mode 2 operation (single-byte transfer modes) are sampled on the rising edge of E. If Tx RQ for one of these channels is asserted when sampled, the DMAC will perform one DMA byte transfer for the channel before sampling the Tx RQ pin of the channel next in the priority. The Tx RQ pins for channels programmed for mode 3 operation (block transfer mode) are sampled on the rising edge of E for the first DMA byte transfer only. If a Tx RQ for one of these channels is asserted when sampled, the first byte of the channel data block is transferred, then the Tx RQ pin is sampled on falling edges of E for subsequent byte transfers (see Figure 6). Once a channel programmed for mode 3 operation begins DMA, that channel has priority of servicing until the channel completes its entire block transfer.

DMA REQUEST 1-2 ($\overline{DRQ1}$, $\overline{DRQ2}$)

These active low TTL-compatible outputs are used by the DMAC to handshake with the MPU in requesting the system bus for DMA operation. $\overline{DRQ1}$ is asserted to indicate that a channel configured for mode 1 operation requires servicing, and $\overline{DRQ2}$ is asserted to indicate that a channel configured for mode 2 or mode 3 operation requires servicing. Once asserted, each output remains asserted until the DMAC completes one DMA byte transfer in mode 1 and mode 2 DMA, or an entire byte block transfer in mode 3 DMA.

DMA GRANT (DGRNT)

This high-impedance input is used to enable MC6844 DMA operation and should be asserted only after the MPU has relinquished the system bus to the DMAC. Typically, DGRNT will be asserted by the MPU in response to a DMA request, indicating that the system bus is available for DMA.

TRANSFER STROBE ($\overline{Tx\ STB}$)

$\overline{Tx\ STB}$ is asserted during each DMA transfer cycle and can be used as a transfer acknowledge for peripheral controllers and as a system VMA. $\overline{Tx\ STB}$ is a TTL-compatible output.

TRANSFER ACKNOWLEDGE A (Tx AKA)

Transfer Acknowledge A is asserted during DMA operation and can be used with Tx AKB to identify the DMA channel being serviced, as shown in Table 1.

CHIP SELECT/TRANSFER ACKNOWLEDGE B ($\overline{CS}$/Tx AKB)

This bidirectional pin serves two functions. During MPU operation it is a chip-select input which when asserted allows MPU access to the DMAC registers. During DMA transfers this pin is for Tx AKB output, used with Tx AKA to identify the DMA channel being serviced (see Table 1).

TABLE 1 — ENCODING ORDER

$\overline{CS}$/Tx AKB	Tx AKA	Channel #
0	0	0
0	1	1
1	0	2
1	1	3

DMAC REGISTERS

All DMAC registers are read/write regsiters, although some of the register status bits are read-only. Table 2 presents a summary of the DMAC control registers, and Table 3 lists address and byte count register addresses.

ADDRESS REGISTERS

Associated with each DMA channel is an address register which stores the 16-bit address to be asserted on the system

TABLE 2 — DMAC CONTROL REGISTERS

Register	Address (Hex)	Register Content							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Control	1x*	DMA End Flag (DEND)	Busy/Ready Flag	Not Used	Not Used	Address $\overline{\text{Up}}$/Down	MCA	MCB	Read/$\overline{\text{Write}}$ (R/$\overline{\text{W}}$)
Priority Control	14	Rotate Control	Not Used	Not Used	Not Used	Request Enable #3 (RE3)	Request Enable #2 (RE2)	Request Enable #1 (RE1)	Request Enable #0 (RE0)
Interrupt Control	15	DEND IRQ Flag	Not Used	Not Used	Not Used	DEND IRQ Enable #3 (DIE3)	DEND IRQ Enable #2 (DIE2)	DEND IRQ Enable #1 (DIE1)	DEND IRQ Enable #0 (DIE0)
Data Chain	16	Not Used	Not Used	Not Used	Not Used	Two/Four Channel Select (2/4)	Data Chain Channel Select B	Data Chain Channel Select A	Data Chain Enable

*The x represents the binary equivalent of the channel desired.

TABLE 3 — ADDRESS AND BYTE COUNT REGISTERS

Register	Channel	Address (Hex)
Address High	0	0
Address Low	0	1
Byte Count High	0	2
Byte Count Low	0	3
Address High	1	4
Address Low	1	5
Byte Count High	1	6
Byte Count Low	1	7
Address High	2	8
Address Low	2	9
Byte Count High	2	A
Byte Count Low	2	B
Address High	3	C
Address Low	3	D
Byte Count High	3	E
Byte Count Low	3	F

address bus during the next DMA cycle of the channel. After each DMA byte transfer, the address register will increment or decrement according to the state of bit 3 of the appropriate channel control register. The starting address of a DMA data block should be stored in the address register of a channel to be used before beginning DMA operation with the channel.

BYTE COUNT REGISTERS

Each channel has a 16-bit byte count register which stores the number of DMA cycles remaining in a channel DMA block. This register should be loaded with the number of bytes to be transferred by a channel before the channel begins DMA. The byte count register is decremented at the beginning of a DMA cycle

CHANNEL CONTROL REGISTERS

A channel control register associated with each channel is used to control the channel mode of operation, the state of the R/$\overline{\text{W}}$ line during DMA, and whether the channel address register will increment or decrement after each DMA cycle. The channel control registers contain two read-only status flags which report the status of the channel. The channel control register bits are defined as follows:

Bit 0 R/$\overline{\text{W}}$	Read/Write. The direction of DMA transfer is determined by the state of this bit When this bit is a "1", R/W will be asserted high by the DMAC during DMA, and memory will be read by the peripheral controller. When this bit is a "0", R/W will be asserted low by the DMAC during DMA and data transfer will be from the peripheral controller to memory
Bit 1 MCB	Mode Control B. This bit is used to select the channel DMA mode. When this bit is a "1", mode 3 operation is selected. When this bit is clear, either mode 1 or mode 2 operation is selected according to the state of channel control register bit 2. Table 4 shows the DMA mode options.

TABLE 4 — DMA MODE SELECT

MCA	MCB	DMA Transfer Mode
0	0	Mode 2
0	1	Mode 3
1	0	Mode 1
1	1	Undefined

Bit 2 MCA	Mode Control A. This bit is used with MCB to select the channel DMA mode. When MCB is set, this bit must be clear and mode 3 operation is selected. Setting both MCA and MCB to a "1" places the DMAC into an undefined mode of operation. With MCB clear, setting MCA to a "1" places the channel into mode 1 and clearing MCA places the channel into mode 2 (see Table 2).
Bit 3	Address Up/Down. Bit 3 controls address register increment/decrement during DMA. If this bit is set to a "1", the address register decrements with each DMA cycle, if it is clear, the address register increments with each DMA cycle
Bits 4-5	Not used
Bit 6	Busy/Ready Flag The Busy/Ready flag is read-only status bit that indicates a DMA block transfer is in progress in the channel. After initializing the channel for a block transfer (address register, byte count register, etc.), this flag sets when Tx RQ is recognized and clears during the last block byte transfer
Bit 7 DEND	DMA End Flag (DEND). The DEND flag is used to indicate when a DMA transfer is complete This flag is set during the transfer of the last byte of a DMA block and is cleared by reading the channel control register. This flag will generate an $\overline{\text{IRQ}}$ interrupt if enabled in the interrupt control register

PRIORITY CONTROL REGISTER

The Priority Control Register is used to individually enable each DMA channel and to select the channel service priority scheme, with bits defined as follows:

Bits 0-3 RE0-3	Request Enable 0-3 Each DMA channel is individually enabled by setting the appropriate RE bit (RE0 for channel 0 etc) in the priority control register. A clear channel RE bit inhibits recognition of Tx RQ for the channel.
Bits 4-6	Not used.
Bit 7	Rotate Control One of two channel service priority schemes can be selected by bit 7. When this bit is "0", the fixed priority of servicing is selected in which channel 0 has highest priority, channel 1 has the next highest priority, channel 2 the next highest priority, and channel 3 the last priority. When this bit is set to a "1", the rotating priority of servicing is selected. Rotating priority is initially the same as fixed priority, in that the lower numbered channels initially have the higher prirorities. However, once a channel is serviced in the rotating priority mode, that channel is given last priority of servicing In this scheme the channel last serviced gets the last priority.

INTERRUPT CONTROL REGISTER

The interrupt control register allows the user to selectively enable each channel $\overline{\text{IRQ}}$ interrupt. When enabled, an IRQ is generated when a DMA block transfer is complete. The interrupt control register also has a flag to indicate that the DMAC $\overline{\text{IRQ}}$ is asserted. Interrupt control register bits are defined as follows:

Bits 0-3 DIE0-3	DEND IRQ Enable. These bits enable individual channel $\overline{\text{IRQ}}$ interrupts when set to "1", and mask these interrupts when cleared. The register bit number is the same as the channel number controlled by the bit. An $\overline{\text{IRQ}}$ is asserted only when a DMA block transfer is completed
Bits 4-6	Not used.
Bit 7	DEND IRQ Flag This read-only bit is set to a "1" when the DMAC $\overline{\text{IRQ}}$ is asserted, indicating the end of a channel block transfer (DEND assertion) with interrupt enabled. This flag is cleared and $\overline{\text{IRQ}}$ is negated by a read of the channel control register of the channel causing the $\overline{\text{IRQ}}$ interrupt

DATA CHAIN REGISTER

Repetitive reading or writing of a block of memory can best be performed using the data chain function This function transfers the contents of the channel 3 address and byte count registers into the respective registers of the channel selected for data chaining. These contents are transferred during the E cycle following the transfer of the last byte of a block by the selected channel The data chain register is defined as follows·

Bit 0 DCE	Data Chain Enable. Data chaining is enabled when this bit is set to a "1". When this bit is clear, data chaining is disabled.
Bit 1-2 DCA/B	Data Chain Select A, B. The state of these two bits determine which channel will be updated when data chaining is enabled, as listed in Table 5
Bit 3	Two/Four Channel Select. The DMAC will operate with either two channels or four channels, depending on the state of this bit. When this bit is set to a "1", the four-channel mode is selected, and all four channels are selectable. When this bit is clear, the two-channel mode is selected and only channels 0 and 1 are selectable.
Bits 4-7	Not used.

TABLE 5 — CHANNEL SELECT

DCB Bit 2	DCA Bit 1	Channel #
0	0	0
0	1	1
1	0	2
1	1	Undefined

APPLICATIONS

The MC6844 DMAC can be interfaced to a wide variety of MPUs, including the Motorola MC68000. This section offers examples of MC6844 interface circuits that can be used as starting points in designing the DMAC into a particular system.

$\overline{\mathrm{IRQ}}$, $\overline{\mathrm{DEND}}$, Tx AK GENERATION

Derivation of $\overline{\mathrm{IRQ}}$ (Interrupt Request), DEND (DMA End), and Tx AK (Transfer Acknowledge) for one, two, and four-channel DMA is shown in Figure 10. IRQ, if enabled, is asserted by the DMA to interrupt the MPU whenever a DMA block transfer is completed. Tx AK is asserted during each DMA cycle and is used to handshake with a peripheral controller each time a DMA byte transfer occurs. DEND is used to handshake with a peripheral controller each time a DMA block transfer is complete.

Each circuit uses DMA GRANT to demultiplex the $\overline{\mathrm{IRQ}/\mathrm{DEND}}$ DMAC output to ensure that the system $\overline{\mathrm{IRQ}}$ is asserted at the proper time, only during MCU operation. Whenever DMA GRANT is high, $\overline{\mathrm{IRQ}}$ is negated.

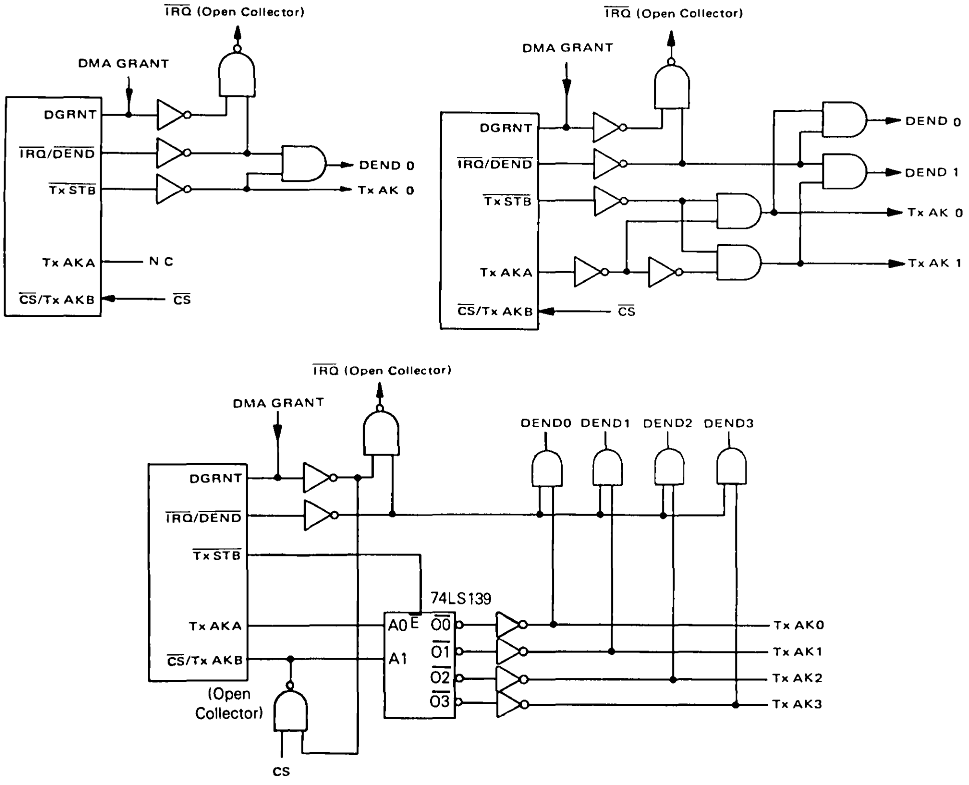
The circuits also generate DEND and Tx AK for the proper channel, gated by Tx STB.

The one-channel DMA mode requires no channel decoding, so for this mode Tx AK is derived from Tx $\overline{\mathrm{STB}}$ directly, and Tx $\overline{\mathrm{STB}}$ is used to demultiplex the $\overline{\mathrm{IRQ}/\mathrm{DEND}}$ output for $\overline{\mathrm{DEND}}$ generation.

The two-channel mode circuit is similar to the one-channel circuit, but uses Tx AKA to identify the active channel and generate the appropriate channel signal (see Table 1)

The four-channel circuit is functionally similar to the two-channel circuit but uses a 74LS139 to decode Tx AKA and Tx AKB for channel identification The DMAC $\overline{\mathrm{CS}}$/Tx AKB pin is bidirectional during four-channel operation, so an open collector gate must be used to drive $\overline{\mathrm{CS}}$ in order to avoid drive contention.

FIGURE 10 — $\overline{\mathrm{IRQ}}$, $\overline{\mathrm{DEND}}$, Tx AK GENERATION


IRQ (Open Collector)
DMA GRANT
DGRNT
IRQ/DEND
Tx STB
Tx AKA
CS/Tx AKB
DEND 0
Tx AK 0
N C
CS
IRQ (Open Collector)
DMA GRANT
DGRNT
IRQ/DEND
Tx STB
Tx AKA
CS/Tx AKB
DEND 0
DEND 1
Tx AK 0
Tx AK 1
CS
IRQ (Open Collector)
DMA GRANT
DGRNT
IRQ/DEND
Tx STB
Tx AKA
CS/Tx AKB
(Open Collector)
CS
74LS139
A0
E
A1
00
01
02
03
DEND0
DEND1
DEND2
DEND3
Tx AK0
Tx AK1
Tx AK2
Tx AK3

MC68000 BUS ARBITRATION INTERFACE

Figure 11 shows an MC6844/MC68000 interface for DMAC mode 2 or mode 3 operation The MC68000 Advanced Information Data Sheet should be consulted for complete understanding of the circuit.

The MC6844 must be initialized for transfer mode, byte count, DMA starting address, etc

Initially DGRNT is low, $\overline{\text{BGACK}}$ output is high, and $\overline{\text{Tx STB}}$ is high The MC6844 responds to a Tx RQ by asserting DRQH Assertion of Tx RQ also asserts MC68000 $\overline{\text{BR}}$ For DMA transfer, two conditions must be met 1) DMAC $\overline{\text{DRQH}}$ must be asserted and 2) all bus masters must relinquish the system bus. Once $\overline{\text{DRQH}}$ is asserted it remains asserted low until DMA byte transfer in the halt-steal mode or until the last byte of a DMA memory block is being transferred in the halt-burst mode A relinquishing of the bus by all bus masters is indicated by negated $\overline{\text{BGACK}}$, $\overline{\text{AS}}$, and $\overline{\text{DTACK}}$ after the MC68000 asserts $\overline{\text{BG}}$ in response to a bus request

When both conditions are met, the NAND flip-flop is set by assertion of LS138 $\overline{\text{O3}}$, asserting DGRNT and $\overline{\text{BGACK}}$. The DMAC then performs a byte transfer in the halt-steal mode or a block of byte transfers in the halt-burst mode

The NAND flip-flop is cleared on the rising edge of $\overline{\text{Tx STB}}$ after asserting during each DMA cycle in the halt-steal mode, and during the last DMA cycle of a DMA block in the halt-burst mode (see MC6844 timing diagrams).

Note that $\overline{\text{BR}}$ to the MC68000 is negated when $\overline{\text{BGACK}}$ is asserted, satisfying an MC68000 requirement

MC6800 BUS ARBITRATION INTERFACE

A typical system design, using the MC6800/MC6844, is shown in Figure 12 A clock generator/driver is used which will stretch the MPU clock during DMA operation while generating a non-stretched clock for system memory Priority logic is used to give highest priority to refresh request, since memory refresh and DMA transfers must not occur during the same E cycles

During mode 2 or 3 DMA operation, the clock generator has no control over DMA Grant. To prevent DMA operation in mode 1 during a memory refresh cycle, system E must be gated with refresh grant DGRNT must be the ORed output of bus available (BA) and DMA grant from the clock generator in order to support all 3 DMA modes of operation.

During the DMA cycle, a system VMA signal must be generated by the DMAC This is done by ORing $\overline{\text{Tx STB}}$ and the MPU VMA line

MC6844/MC6809 BUS ARBITRATION INTERFACE

An MC6844/MC6809 interface is presented in Figure 13 This circuit ensures that MC6809 $\overline{\text{DMA/BREQ}}$ is asserted only during Q high, an MC6809 requirement The circuit will also generate a system VMA (valid memory address), often referred to as DMA $\overline{\text{VMA}}$.

The MC6809 does not generate a $\overline{\text{VMA}}$ output since the only invalid address asserted by the MPU is \$FFFF with R/W asserted high Therefore, an MC6809 system does not normally need a VMA circuit When using the MC6844 for DMA in an MC6809 system, however, a $\overline{\text{VMA}}$ circuit is required since the address lines are floating during dead cycles between the MPU and DMA modes. Devices on the bus must be deselected during this time

Initially, in the MPU mode, $\overline{\text{DRQ1/2}}$ is negated (high level), and the Q output of U3 is high. The output of the exclusive OR gate U4 is therefore a low, inhibiting clocking of U3 by forcing the output of U5 to remain a low. When $\overline{\text{DRQ1/2}}$ is asserted low, the output of U4 changes to a high If the MC6809 Q output is high at this time, the output of U5 changes to a high, clocking U3. If the MC6809 Q output is low at this time, the output of U5 will be driven high on the next rising edge of Q, clocking U3. When U3 is clocked, the Q output of U3 changes to a low asserting MC6809 $\overline{\text{DMA/BREQ}}$ The output of U4 at this time is a low, since both of the U4 inputs are low

FIGURE 11 — MC68000/MC6844 INTERFACE

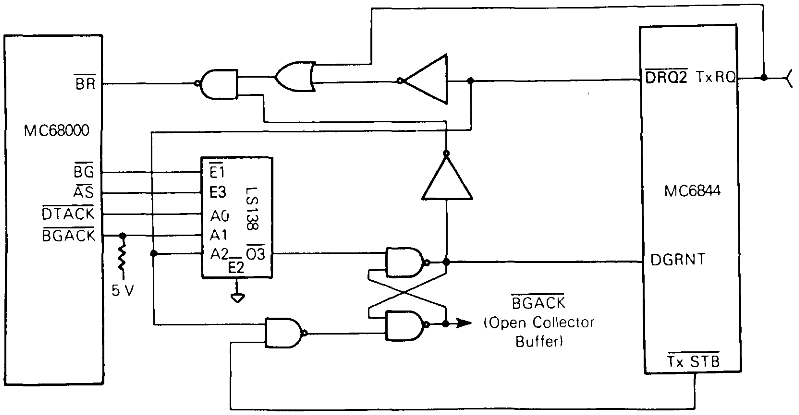


FIGURE 12 — MC6800/MC6844 INTERFACE

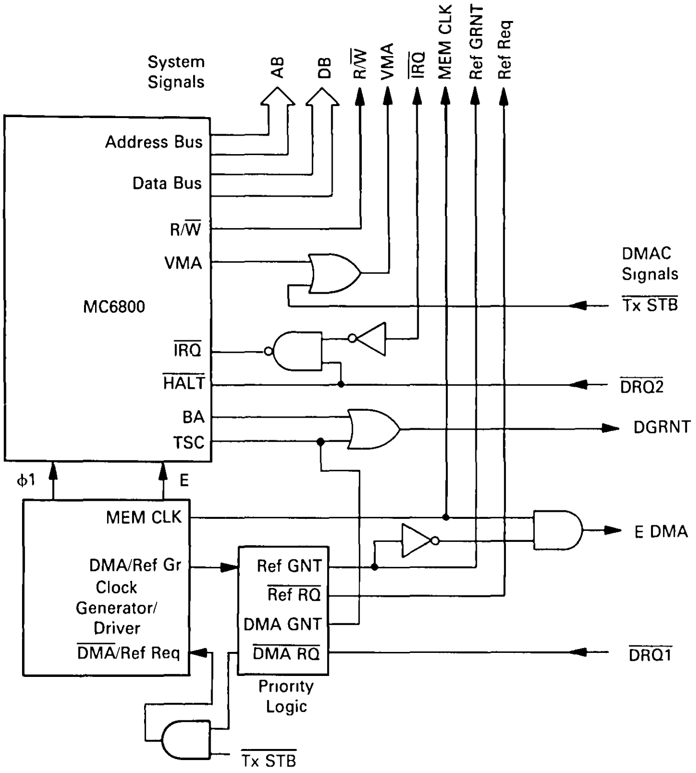
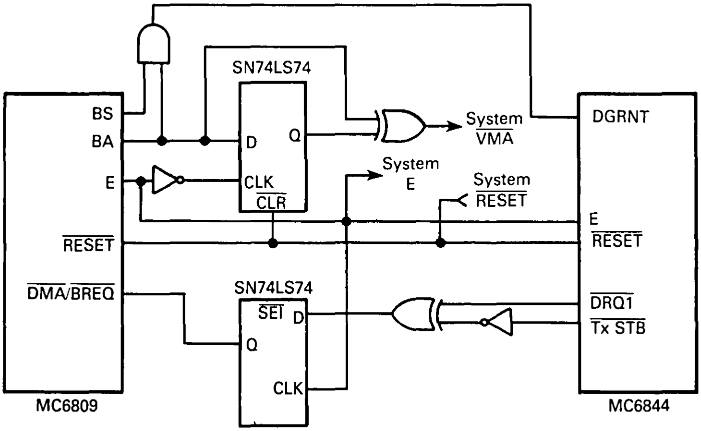

System Signals
AB
DB
R/W̄
VMA
IRQ̄
MEM CLK
Ref GRNT
Ref Req
Address Bus
Data Bus
R/W̄
VMA
MC6800
IRQ̄
HALT̄
BA
TSC
DMAC Signals
Tx STB̄
DRQ2̄
DGRNT
φ1
E
MEM CLK
DMA/Ref Gr
Clock Generator/ Driver
DMA̅/Ref Req
E DMA
Ref GNT
Ref RQ̄
DMA GNT
DMA RQ̄
DRQ1̄
Priority Logic
Tx STB̄

FIGURE 13 — MC6844/MC6809 INTEFACE FOR MODE 1 DMA


SN74LS74
BS
BA
E
D
Q
CLK
CLR̄
System VMA̅
System E
System RESET̄
DGRNT
E
RESET̄
RESET̄
DMA̅/BREQ̄
SN74LS74
SET̄
D
Q
CLK
DRQ1̄
Tx STB̄
MC6809
MC6844

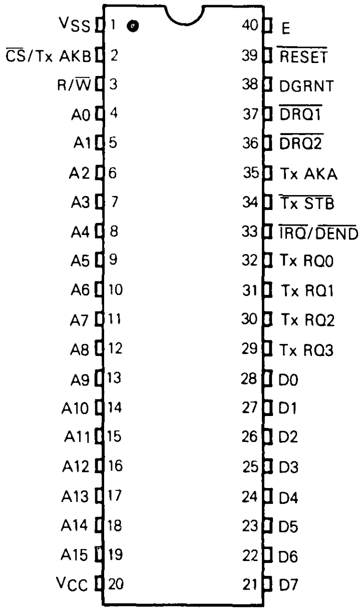
After the DMA transfer, DRQ1/2 is negated by the MC6844, forcing the output of U4 to a high. Once again, U3 will be clocked only when the MC6809 Q output is high.

VMA is generated by U1 and U2 Initially, in the MPU mode, U1 is clear, with a low Q output. The BA (bus available) output of the MC6809 is also a low. Therefore, the output of U2 ($\overline{\text{VMA}}$) is low ($\overline{\text{VMA}}$ asserted) When the MC6809 asserts BA for DMA, the output of U2 becomes high, indicating that the address on the system address bus is invalid during this dead cycle between MPU and DMA modes. On the next falling edge of E, U1 is clocked high forcing the output of U2 low during this DMA cycle When BA is negated after DMA, the output of U2 is forced high until the next falling edge of E, indicating invalid address during this dead cycle.

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip	1.0	0°C to 70°C	MC6844S
S Suffix	1.0	−40°C to +85°C	MC6844CS
	1.5	0°C to 70°C	MC68A44S
	1.5	−40°C to +85°C	MC68A44CS
	2.0	0°C to 70°C	MC68B44S
Plastic	1.0	0°C to 70°C	MC6844P
P Suffix	1.0	−40°C to +85°C	MC6844CP
	1.5	0°C to 70°C	MC68A44P
	1.5	−40°C to +85°C	MC68A44CP
	2.0	0°C to 70°C	MC68B44P

PIN ASSIGNMENT


VSS 1
CS/Tx AKB 2
R/W 3
A0 4
A1 5
A2 6
A3 7
A4 8
A5 9
A6 10
A7 11
A8 12
A9 13
A10 14
A11 15
A12 16
A13 17
A14 18
A15 19
VCC 20
40 E
39 RESET
38 DGRNT
37 DRQ1
36 DRQ2
35 Tx AKA
34 Tx STB
33 IRQ/DEND
32 Tx RQ0
31 Tx RQ1
30 Tx RQ2
29 Tx RQ3
28 D0
27 D1
26 D2
25 D3
24 D4
23 D5
22 D6
21 D7

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC6845

CRT Controller (CRTC)

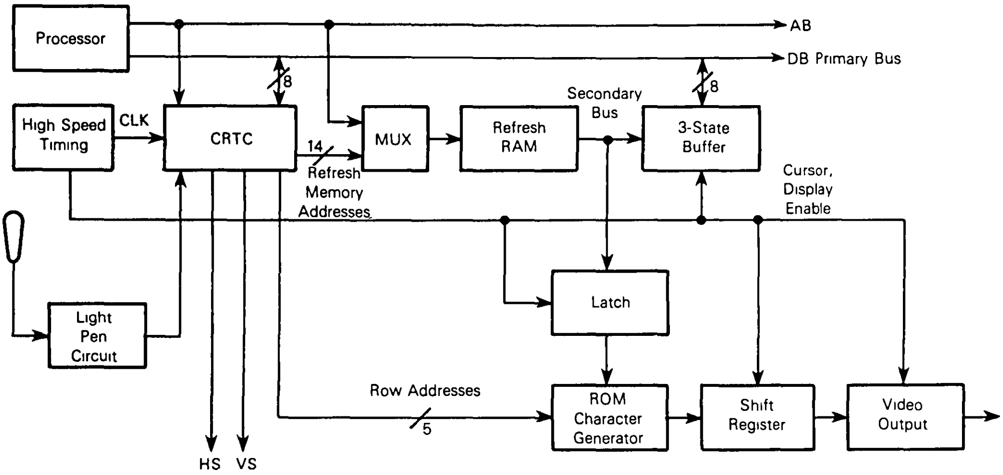
The MC6845 CRT controller performs the interface between an MPU and a raster-scan CRT display. It is intended for use in MPU-based controllers for CRT terminals in stand-alone or cluster configurations.

The CRTC is optimized for the hardware/software balance required for maximum flexibility. All keyboard functions, reads, writes, cursor movements, and editing are under processor control. The CRTC provides video timing and refresh memory addressing.

- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games; Information Displays
- Alphanumeric, Semi-Graphic, and Full-Graphic Capability
- Fully Programmable Via Processor Data Bus. Timing May Be Generated for Almost Any Alphanumeric Screen Format, e.g., 80 × 24, 72 × 64, 132 × 20
- Single +5 V Supply
- M6800 Compatible Bus Interface
- TTL-Compatible Inputs and Outputs
- Start Address Register Provides Hardware Scroll (by Page or Character)
- Programmable Cursor Register Allows control of Cursor Format and Blink Rate
- Light Pen Register
- Refresh (Screen) Memory May be Multiplexed Between the CRTC and the MPU Thus Removing the Requirements for Line Buffers or External DMA Devices
- Programmable Interlace or Non-Interlace Scan Modes
- 14-Bit Refresh Address Allows Up to 16K of Refresh Memory for Use in Character or Semi-Graphic Displays
- 5-Bit Row Address Allows Up to 32 Scan-Line Character Blocks
- By Utilizing Both the Refresh Addresses and the Row Addresses, a 512K Address Space is Available for Use in Graphics Systems
- Refresh Addresses are Provided During Retrace, Allowing the CRTC to Provide Row Addresses to Refresh Dynamics RAMs
- Pin Compatible with the MC6835

This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — TYPICAL CRT CONTROLLER APPLICATION


Processor
AB
DB Primary Bus
8
High Speed Timing
CLK
CRTC
14
Refresh Memory Addresses
MUX
Refresh RAM
Secondary Bus
8
3-State Buffer
Cursor, Display Enable
Light Pen Circuit
Latch
Row Addresses
5
ROM Character Generator
Shift Register
Video Output
HS
VS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6845, MC68A45, MC68B45 MC6845C, MC68A45C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

The device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Package Cerdip Package	θ_{JA}	 100 60	°C/W

RECOMMENDED OPERATING CONDITIONS

Characteristics	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K - (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part, K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted, see Figures 2-4)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	−0.3	–	0.8	V
Input Leakage Current		I_{in}	–	0.1	2.5	μA
Hi-Z State Input Current (V_{CC} = 5.25 V) (V_{in} = 0.4 to 2.4 V)		I_{TSI}	−10	–	10	μA
Output High Voltage (I_{Load} = −205 μA) (I_{Load} = −100 μA)	 D0-D7 Other Outputs	V_{OH}	 2.4 2.4	 3.0 3.0	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA)		V_{OL}	–	0.3	0.4	V
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	600	750	mW
Input Capacitance	D0-D7 All Others	C_{in}	– –	– –	12.5 10	pF
Output Capacitance	All Outputs	C_{out}	–	–	10	pF

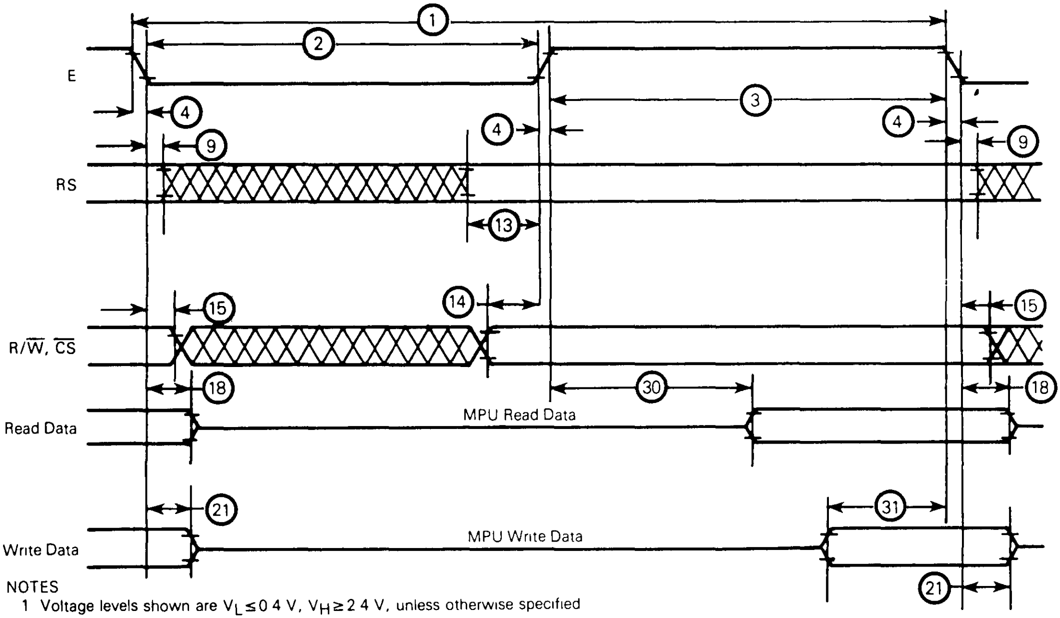
BUS TIMING CHARACTERISTICS (See Notes 1 and 2) (Reference Figures 2 and 3)

Ident. Number	Characteristic	Symbol	MC6845		MC68A45		MC68B45		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10**	0.67	10	0.5	10**	µs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time (RS)	t_{AH}	10	–	10	–	10	–	ns
13	RS Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	R/$\overline{W}$ and $\overline{CS}$ Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	R/$\overline{W}$ and $\overline{CS}$ Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	0	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high impedance)

**The E clock may be low for extended periods provided the CLK input is active

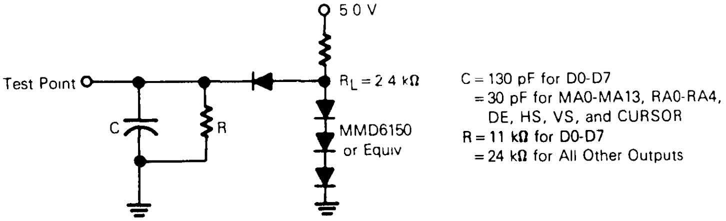
FIGURE 2 — MC6845 BUS TIMING


E
RS
R/$\overline{W}$, $\overline{CS}$
Read Data
Write Data
①
②
③
④
⑨
⑬
⑭
⑮
⑱
㉑
㉚
㉛
MPU Read Data
MPU Write Data

NOTES

1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

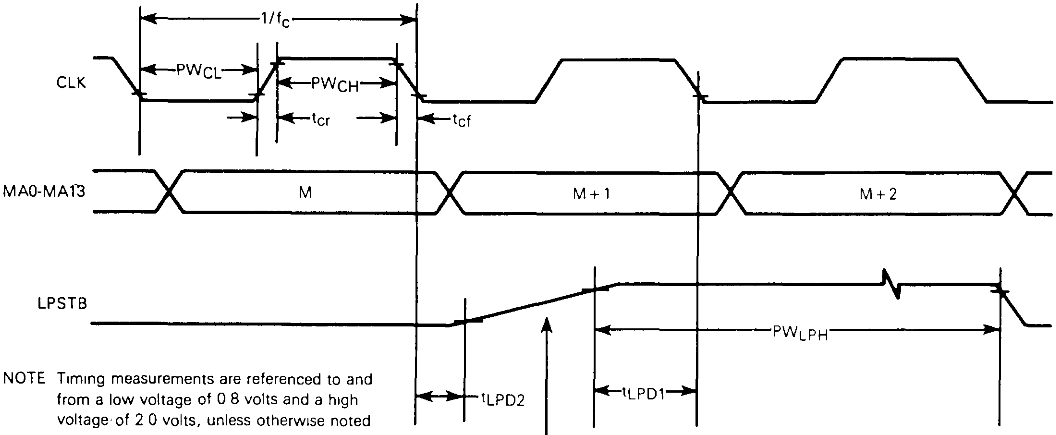
FIGURE 3 — BUS TIMING TEST LOAD


5.0 V
Test Point
$R_L = 2.4$ kΩ
C
R
MMD6150 or Equiv
C = 130 pF for D0-D7
= 30 pF for MA0-MA13, RA0-RA4, DE, HS, VS, and CURSOR
R = 11 kΩ for D0-D7
= 24 kΩ for All Other Outputs

CRTC TIMING CHARACTERISTICS (Reference Figures 4 and 5)

Characteristic	Symbol	Min	Max	Unit
Minimum Clock Pulse Width, Low	P_{WCL}	150	–	ns
Minimum Clock Pulse Width, High	P_{WCH}	150	–	ns
Clock Frequency	f_c	–	3.0	MHz
Rise and Fall Time for Clock Input	t_{cr}, t_{cf}	–	20	ns
Memory Address Delay Time	t_{MAD}	–	160	ns
Raster Address Delay Time	t_{RAD}	–	160	ns
Display Timing Delay Time	t_{DTD}	–	250	ns
Horizontal Sync Delay Time	t_{HSD}	–	250	ns

FIGURE 5 — CRTC-CLK, MA0-MA13, AND LPSTB TIMING DIAGRAM



NOTE Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

When the CRTC detects the rising edge of LPSTB in this period, the CRTC sets the Refresh Memory Address 'M+2' into the LIGHT PEN REGISTER.

tLPD1, tLPD2 Period of uncertainty for the Refresh Memory Address

CRTC INTERFACE SYSTEM DESCRIPTION

The CRT controller generates the signals necessary to interface a digital system to a raster scan CRT display. In this type of display, an electron beam starts in the upper left hand corner, moves quickly across the screen and returns. This action is called a horizontal scan. After each horizontal scan the beam is incrementally moved down in the vertical direction until it has reached the bottom. At this point one frame has been displayed, as the beam has made many horizontal scans and one vertical scan.

Two types of raster scanning are used in CRTs, interlace and non-interlace, shown in Figures 6 and 7. Non-interlace scanning consists of one field per frame. The scan lines in Figure 6 are shown as solid lines and the retrace patterns are indicated by the dotted lines. Increasing the number of frames per second will decrease the flicker. Ordinarily, either a 50 or 60 frame per second refresh rate is used to minimize beating between the CRT and the power line frequency. This prevents the displayed data from weaving.

Interlace scanning is used in broadcast TV and on data monitors where high density or high resolution data must be displayed. Two fields, or vertical scans are made down the screen for each single picture or frame. The first field (even field) starts in the upper left hand corner, the second (odd field) in the upper center. Both fields overlap as shown in Figure 7, thus interlacing the two fields into a single frame.

In order to display the characters on the CRT screen the frames must be continually repeated. The data to be displayed is stored in the refresh (screen) memory by the MPU controlling the data processing system. The data is usually written in ASCII code, so it cannot be directly displayed as characters. A character generator ROM is typically used to convert the ASCII codes into the "dot" pattern for every character.

The most common method of generating characters is to create a matrix of dots "x" dots (columns) wide and "y" dots (rows) high. Each character is created by selectively filling in

FIGURE 6 — RASTER SCAN SYSTEM (NON-INTERLACE)

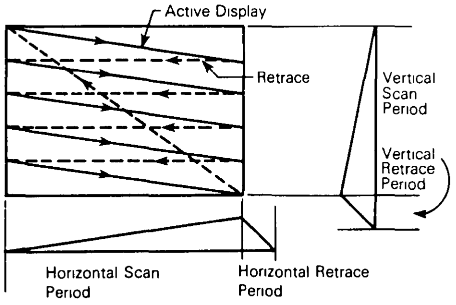
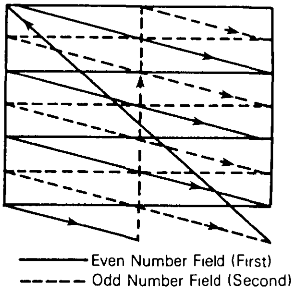


FIGURE 7 — RASTER SCAN SYSTEM (INTERLACE)


—— Even Number Field (First)
- - - - Odd Number Field (Second)

the dots. As "x" and "y" get larger a more detailed character may be created Two common dot matrices are 5×7 and 7×9. Many variations of these standards will allow Chinese, Japanese, or Arabic letters instead of English Since characters require some space between them, a character block larger than the character is typically used, as shown in Figure 8 The figure also shows the corresponding timing and levels for a video signal that would generate the characters

Referring to Figure 1, the CRT controller generates the refresh addresses (MA0-MA13), row addresses (RA0-RA4), and the video timing (vertical sync — VS, horizontal sync — HS, and display enable — DE) Other functions include an internal cursor register which generates a cursor output when its contents compare to the current refresh address A light pen strobe input signal allows capture of the refresh address in an internal light pen register

All timing in the CRTC is derived from the CLK input In alphanumeric terminals, this signal is the character rate The video rate or "dot" clock is externally divided by high-speed logic (TTL) to generate the CLK input The high-speed logic must also generate the timing and control signals necessary for the shift register, latch, and MUX control

The processor communicates with the CRTC through an 8-bit data bus by reading or writing into the 19 registers

The refresh memory address is multiplexed between the processor and the CRTC Data appears on a secondary bus separate from the processor's bus The secondary data bus concept in no way precludes using the refresh RAM for other purposes It looks like any other RAM to the processor. A number of approaches are possible for solving contentions for the refresh memory.

1. Processor always gets priority (Generally, "hash" occurs as MPU and CRTC clocks are not synchronized)
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times
3. Synchronize the processor with memory wait cycles (states)
4. Synchronize the processor to the character rate as shown in Figure 9 The M6800 processor family works works very well in this configuration as constant cycle lengths are present This method provides no overhead for the processor as there is never a contention for a memory access All accesses are transparent

FIGURE 8 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL

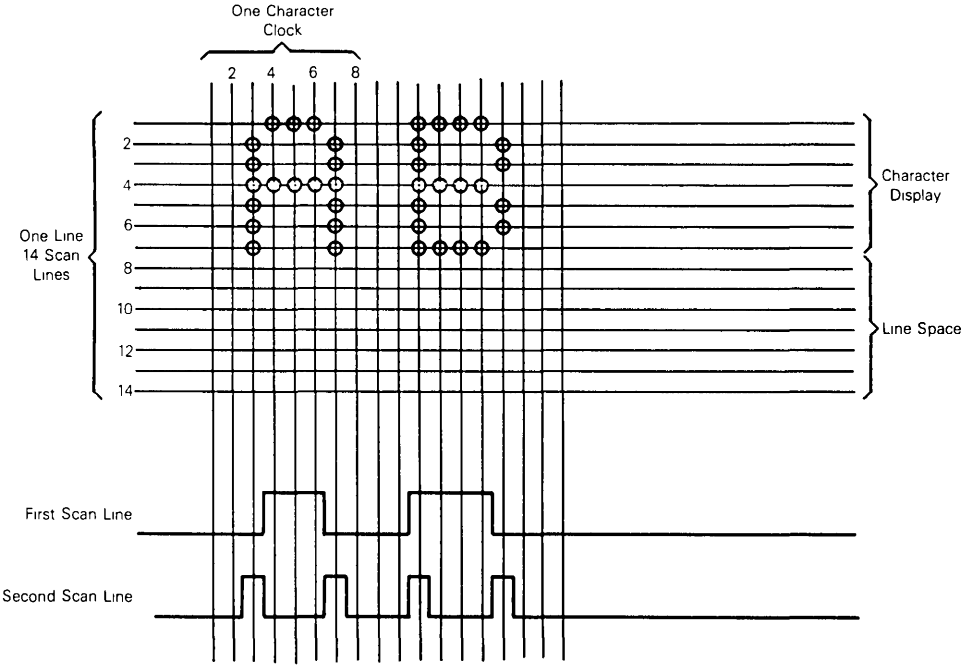
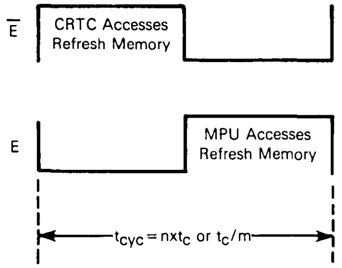


FIGURE 9 — TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING M6800 FAMILY MPU


$\overline{E}$
CRTC Accesses Refresh Memory
E
MPU Accesses Refresh Memory
$t_{cyc} = n \times t_C$ or t_C/m

Where m, n are integers, t_C is character period

PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{CS}$, RS, E, and R/$\overline{W}$ for control signals.

Data Bus (D0-D7) — The bidirectional data lines (D0-D7) allow data transfers between the internal CRTC register file and the processor Data bus output drivers are in the high-impedance state until the processor performs a CRTC read operation

Enable (E) — The enable signal is a high-impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC This signal is usually derived from the processor clock The high-to-low transition is the active edge

Chip Select ($\overline{CS}$) — The $\overline{CS}$ line is a high-impedance TTL/MOS compatible input which selects the CRTC, when low, to read or write to the internal register file. This signal should only be active when there is a valid stable address being decoded from the processor

Register Select (RS) — The RS line is a high-impedance TTL/MOS compatible input which selects either the address register (RS = 0) or one of the data register (RS = 1) or the internal register file

Read/Write (R/$\overline{W}$) — The R/$\overline{W}$ line is a high-impedance TTL/MOS compatible input which determines whether the internal register file gets written or read A write is defined as a low level

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and display enable (DE) signals

NOTE

Care should be exercised when interfacing to CRT monitors, as many monitors claiming to be "TTL compatible" have transistor input circuits which require the CRTC or TTL devices buffering signals from the CRTC/video circuits to exceed the maximum-rated drive currents.

Vertical Sync (VS) and Horizontal Sync (HS) — These TTL-compatible outputs are active high signals which drive the monitor directly or are fed to the video processing circuitry to generate a composite video signal The VS signal determines the vertical position of the displayed text while the HS signal determines the horizontal position of the displayed text

Display Enable (DE) — This TTL-compatible output is an active high signal which indicates the CRTC is providing addressing in the active display area

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides memory addresses (MA0-MA13) to scan the refresh RAM Row addresses (RA0-RA4) are also provided for use with character generator ROMs In a graphics system, both the memory addresses and the row addresses would be used to scan the refresh RAM . Both the memory addresses and the row addresses continue to run during vertical retrace thus allowing the CRTC to provide the refresh addresses required to refresh dynamic RAMs

Refresh Memory Addresses (MA0-MA13) — These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory These outputs are capable of driving one standard TTL load and 30 pF

Row Addresses (RA0-RA4) — These five outputs from the internal row address counter are used to address the character generator ROM These outputs are capable of driving one standard TTL load and 30 pF

OTHER PINS

Cursor — This TTL-compatible output indicates a valid cursor address to external video processing logic It is an active high signal.

Clock (CLK) — The CLK is a TTL/MOS-compatible input used to synchronize all CRT functions except for the processor interface An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high-to-low

Light Pen Strobe (LPSTB) — A low-to-high transition on this high-impedance TTL/MOS-compatible input latches the current Refresh Address in the light pen register. The latching of the refresh address is internally synchronized to the character clock (CLK).

V_{CC} and V_{SS} — These inputs supply +5 Vdc ±5% to the CRTC

$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input is used to reset the CRTC. A low level on the $\overline{\text{RESET}}$ input forces the CRTC into the following state:

(a) All counters in the CRTC are cleared and the device stops the display operation.

(b) All the outputs are driven low.

NOTE

The horizontal sync output is not defined until after R2 is programmed.

(c) The control registers of the CRTC are not affected and remain unchanged.

Functionality of $\overline{\text{RESET}}$ differs from that of other M6800 parts in the following functions.

(a) The $\overline{\text{RESET}}$ input and the LPSTB input are encoded as shown in Table 1.

TABLE 1 — CRTC OPERATING MODE

$\overline{\text{RESET}}$	LPSTB	Operating Mode
0	0	Reset
0	1	Test Mode
1	0	Normal Mode
1	1	Normal Mode

The test mode configures the memory addresses as two independent 7-bit counters to minimize test time

(b) After $\overline{\text{RESET}}$ has gone low and (LPSTB = 0), MA0-MA13 and RA0-RA4 will be driven low on the falling edge of CLK. $\overline{\text{RESET}}$ must remain low for at least one cycle of the character clock (CLK)

(c) The CRTC resumes the display operation immediately after the release of $\overline{\text{RESET}}$ DE and the CURSOR are not active until after the first frame has been displayed

CRTC DESCRIPTION

The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus A block diagram of the CRTC is shown in Figure 10

All CRTC timing is derived from the CLK, usually the output of an external dot rate counter Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17 For horizontal timing generation, comparisons result in 1) horizontal sync pulse (HS) of a frequency, position, and width determined by the registers; 2) horizontal display signal of a frequency, position, and duration determined by the registers.

The horizontal counter produces H clock which drives the scan line counter and vertical control. The contents of the raster counter are continuously compared to the maximum scan line address register A coincidence resets the raster counter and clocks the vertical counter.

Comparisons of vertical counter contents and vertical registers result in 1) vertical sync pulse (VS) of a frequency and position determined by the registers; 2) vertical display of a frequency and position determined by the registers

The vertical control logic has other functions

1. Generate row selects, RA0-RA4, from the raster count for the corresponding interlace or non-interlace modes
2. Extend the number of scan lines in the vertical total by the amount programmed in the vertical total adjust register

The linear address generator is driven by the CLK and locates the relative positions of characters in memory with their positions on the screen Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, eight pages of 2K characters, etc Using the start address register, hardware scrolling through 16K characters is possible The linear address generator repeats the same sequence of addresses for each scan line of a character row

The cursor logic determines the cursor location, size, and blink rate on the screen All are programmable

The light pen strobe going high causes the current contents of the address counter to be latched in the light pen register The contents of the light pen register are subsequently read by the processor

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals — R/$\overline{\text{W}}$, $\overline{\text{CS}}$, RS, and E.

REGISTER FILE DESCRIPTIONS

The nineteen registers of the CRTC may be accessed through the data bus Only two memory locations are required as one location is used as a pointer to address one of the remaining eighteen registers These eighteen registers control horizontal timing, vertical timing, interlace operation, row address operation, and define the cursor, cursor address, start address, and light pen register. The register addresses and sizes are shown in Table 2

ADDRESS REGISTER

The address register is a 5-bit write-only register used as an "indirect"or "pointer" register. It contains the address of one of the other eighteen registers. When both RS and $\overline{\text{CS}}$ are low, the address register is selected. When $\overline{\text{CS}}$ is low and RS is high, the register pointed to by the address register is selected

TIMING REGISTERS R0-R9

Figure 11 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left-most displayed character position. Horizontal registers are programmed in character clock time units with respect to the reference as shown in Figure 12 The point of reference for the vertical registers is the top character position displayed. Vertical registers are programmed in scan line times with respect to the reference as shown in Figure 13

Horizontal Total Register (R0) — This 8-bit write-only register determines the horizontal sync (HS) frequency by defining the HS period in character times It is the total of the displayed characters plus the non-displayed character times (retrace) minus one

*FIGURE 10 — CRTC BLOCK DIAGRAM

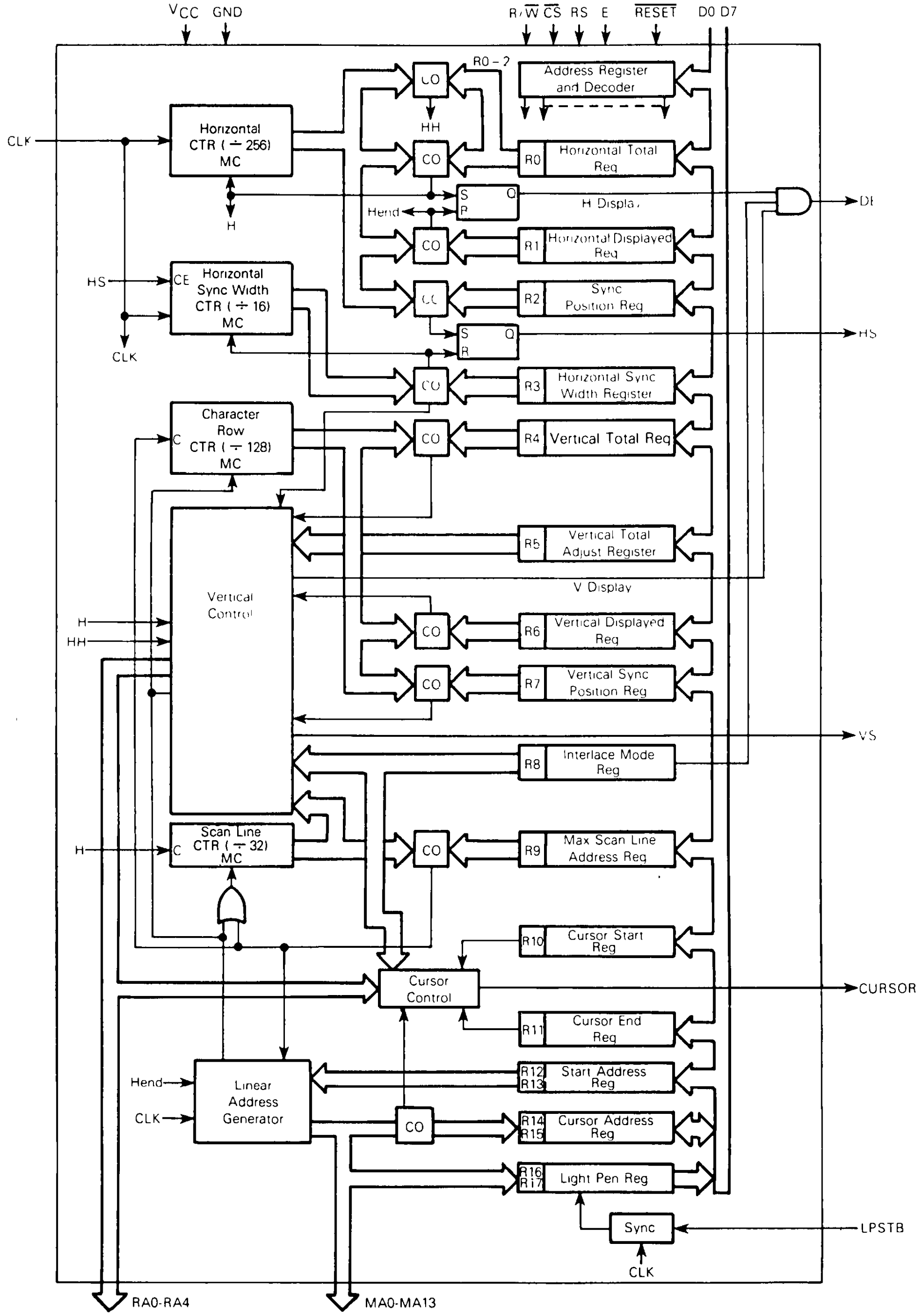
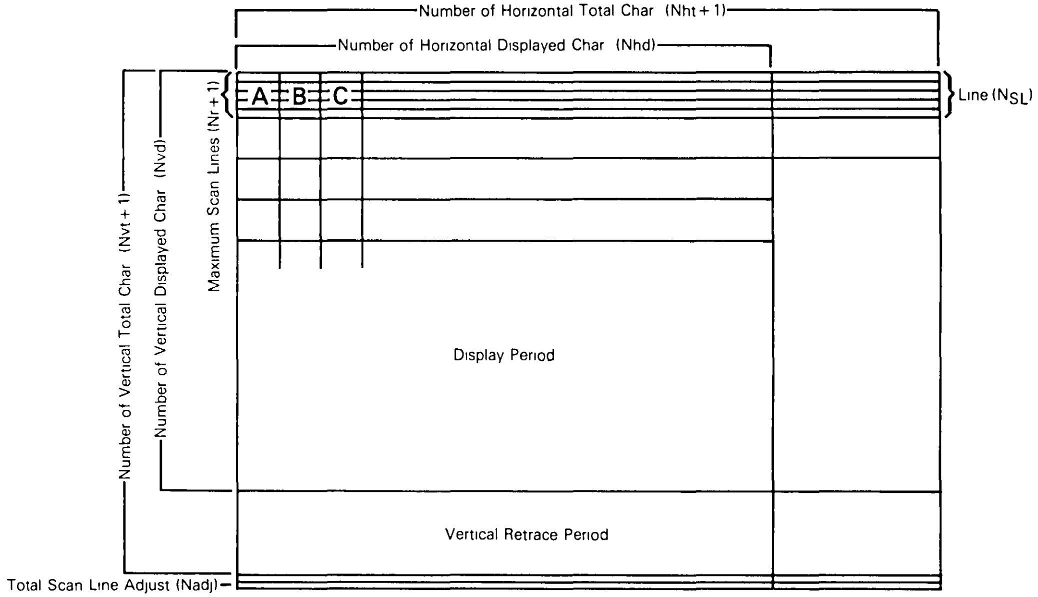

VCC
GND
R/W̄
C̄S̄
RS
E
RESET
D0 D7
R0 – 2
CO
Address Register and Decoder
HH
CLK
Horizontal CTR (÷ 256) MC
R0
Horizontal Total Reg
S
Q
H Display
DE
Hend
R
H
R1
Horizontal Displayed Reg
HS
CE
Horizontal Sync Width CTR (÷ 16) MC
R2
Sync Position Reg
HS
CLK
R3
Horizontal Sync Width Register
Character Row CTR (÷ 128) MC
C
R4
Vertical Total Req
R5
Vertical Total Adjust Register
V Display
Vertical Control
H
HH
R6
Vertical Displayed Reg
R7
Vertical Sync Position Reg
VS
R8
Interlace Mode Reg
Scan Line CTR (÷ 32) MC
H
R9
Max Scan Line Address Reg
R10
Cursor Start Reg
Cursor Control
CURSOR
R11
Cursor End Reg
Hend
CLK
Linear Address Generator
R12
R13
Start Address Reg
R14
R15
Cursor Address Reg
R16
R17
Light Pen Reg
Sync
LPSTB
CLK
RA0-RA4
MA0-MA13

TABLE 2 — CRTC INTERNAL REGISTER ASSIGNMENT

CS̄	RS	Address Register					Register #	Register File	Program Unit	Read	Write	Number of Bits							
		4	3	2	1	0						7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X	–	–	–	–	╲	╲	╲	╲	╲	╲	╲	╲
0	0	X	X	X	X	X	AR	Address Register	–	No	Yes	╲	╲	╲					
0	1	0	0	0	0	0	R0	Horizontal Total	Char	No	Yes								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Char	No	Yes								
0	1	0	0	0	1	0	R2	H Sync Position	Char	No	Yes								
0	1	0	0	0	1	1	R3	Sync Width	–	No	Yes	╲	╲	╲	╲	H	H	H	H
0	1	0	0	1	0	0	R4	Vertical Total	Char Row	No	Yes	╲							
0	1	0	0	1	0	1	R5	V Total Adjust	Scan Line	No	Yes	╲	╲	╲					
0	1	0	0	1	1	0	R6	Vertical Displayed	Char Row	No	Yes	╲							
0	1	0	0	1	1	1	R7	V Sync Position	Char Row	No	Yes	╲							
0	1	0	1	0	0	0	R8	Interlace Mode and Skew	Note 1	No	Yes	╲	╲	╲	╲	╲	╲	I	I
0	1	0	1	0	0	1	R9	Max Scan Line Address	Scan Line	No	Yes	╲	╲	╲					
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line	No	Yes	╲	B	P			(Note 2)		
0	1	0	1	0	1	1	R11	Cursor End	Scan Line	No	Yes	╲	╲	╲					
0	1	0	1	1	0	0	R12	Start Address (H)	–	No	Yes	0	0						
0	1	0	1	1	0	1	R13	Start Address (L)	–	No	Yes								
0	1	0	1	1	1	0	R14	Cursor (H)	–	Yes	Yes	0	0						
0	1	0	1	1	1	1	R15	Cursor (L)	–	Yes	Yes								
0	1	1	0	0	0	0	R16	Light Pen (H)	–	Yes	No	0	0						
0	1	1	0	0	0	1	R17	Light Pen (L)	–	Yes	No								

NOTES
1 The interlace is shown in Table 3
2 Bit 5 of the cursor start raster register is used for blink period control, and bit 6 is used to select blink or no-blink

FIGURE 11 — ILLUSTRATION OF THE CRT SCREEN FORMAT


Number of Horizontal Total Char (Nht + 1)
Number of Horizontal Displayed Char (Nhd)
A
B
C
Line (N$_{SL}$)
Number of Vertical Total Char (Nvt + 1)
Number of Vertical Displayed Char (Nvd)
Maximum Scan Lines (Nr + 1)
Display Period
Vertical Retrace Period
Total Scan Line Adjust (Nadj)

NOTE 1 Timing values are described in Table 5

Horizontal Displayed Register (R1) — This 8-bit write-only register determines the number of displayed characters per line. Any 8-bit number may be programmed as long as the contents of R0 are greater than the contents of R1.

Horizontal Sync Position Register (R2) — This 8-bit write-only register controls the HS position The horizontal sync position defines the horizontal sync delay (front porch) and the horizontal scan delay (back porch) When the programmed value of this register is increased, the display on the CRT screen is shifted to the left When the programmed value is decreased the display is shifted to the right Any 8-bit number may be programmed as long as the sum of the contents of R2 and R3 are less than the contents of R0 R2 must be greater than R1.

Sync Width Register (R3) — This 8-bit write-only register determines the width of the horizontal sync (HS) pulse The vertical sync pulse width is fixed at 16 scan-line times

The HS pulse width may be programmed from 1-to-15 character clock periods thus allowing compatibility with the HS pulse width specifications of many different monitors If zero is written into this register then no HS is provided.

Horizontal Timing Summary (Figure 12) — The difference between R0 and R1 is the horizontal blanking interval. This interval in the horizontal scan period allows the beam to return (retrace) to the left side of the screen. The retrace time is determined by the monitor's horizontal scan components Retrace time is less than the horizontal blanking interval A good rule of thumb is to make the horizontal blanking about 20% of the total horizontal scanning period for a CRT. In inexpensive TV receivers, the beam overscans the display screen so that aging of parts does not result in underscanning. Because of this, the retrace time should be about one third the horizontal scanning period The horizontal sync delay, HS pulse width, and horizontal scan delay are typically programmed with a 1 2.2 ratio

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) — The frequency of VS is determined by both R4 and R5 The calculated number of character row times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate The integer number of character row times minus one is programmed in the 7-bit write-only vertical total register (R4) The fraction of character line times is programmed in the 5-bit write-only vertical total adjust register (R5) as the number of scan lines required

Vertical Displayed Register (R6) — This 7-bit write-only register specifies the number of displayed character rows on the CRT screen, and is programmed in character row times Any number smaller than the contents of R4 may be programmed into R6.

Vertical Sync Position (R7) — This 7-bit write-only register controls the position of vertical sync with respect to the reference It is programmed in character row times When the programmed value of this register is increased, the display position of the CRT screen is shifted up When the programmed value is decreased the display position is shifted down Any number equal to or less than the vertical total (R4) and greater than or equal to the vertical displayed (R6) may be used

Interlace Mode and Skew Register (R8) — The MC6845 only allows control of the interlace modes as programmed by the low order two bits of this write-only register. Table 3 shows the interlace modes available to the user These modes are selected using the two low order bits of this 6-bit write-only register

TABLE 3 — INTERLACE MODE REGISTER

Bit 1	Bit 0	Mode
0	0	Normal Sync Mode (Non-Interlace)
1	0	
0	1	Interlace Sync Mode
1	1	Interlace Sync and Video Mode

In the normal sync mode (non-interlace) only one field is available as shown in Figures 6 and 14a Each scan line is refreshed at the VS frequency (e g , 50 or 60 Hz)

Two interlace modes are available as shown in Figures 7, 14b, and 14c The frame time is divided between even and odd alternating fields The horizontal and vertical timing relationship (VS delayed by one half scan line time) results in the displacement of scan lines in the odd field with respect to the even field

In the interlace sync mode the same information is painted in both fields as shown in Figure 14b This is a useful mode for filling in a character to enhance readability

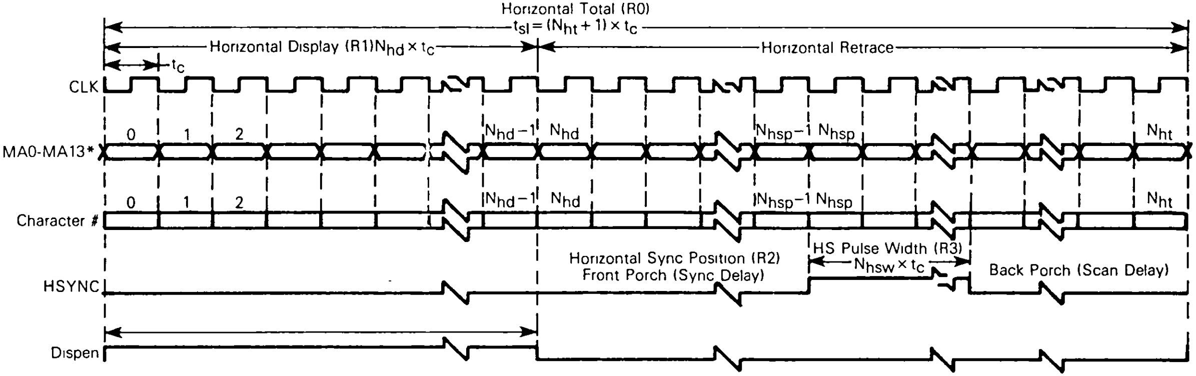
In the interlace sync and video mode, shown in Figure 14c, alternating lines of the character are displayed in the even field and the odd field. This effectively doubles the given bandwidth of the CRT monitor

Care must be taken when using either interlace mode to avoid an apparent flicker effect This flicker effect is due to the doubling of the refresh time for all scan lines since each field is displayed alternately and may be minimized with proper monitor design (e g , longer persistence phosphors)

In addition, there are restrictions on the programming of the CRTC registers for interlace operation

1 The horizontal total register value, R0, must be odd (i e , an even number of character times)
2 For interlace sync and video mode only, the maximum scan-line address, R9, must be odd (i e , an even number of scan lines)
3 For interlace sync and video mode only, the number (Nvd) programmed into the vertical display register (R6) must be one half the actual number required The even numbered scan lines are displayed in the even field and the odd numbered scan lines are displayed in the odd field
4 For interlace sync and video mode only, the cursor start register (R10) and cursor end register (R11) must both be even or both odd depending on which field the cursor is to be displayed in A full block cursor will be displayed in both the even and the odd field when the cursor end register (R11) is programmed to a value greater than the value in the maximum scan line address register (R9)

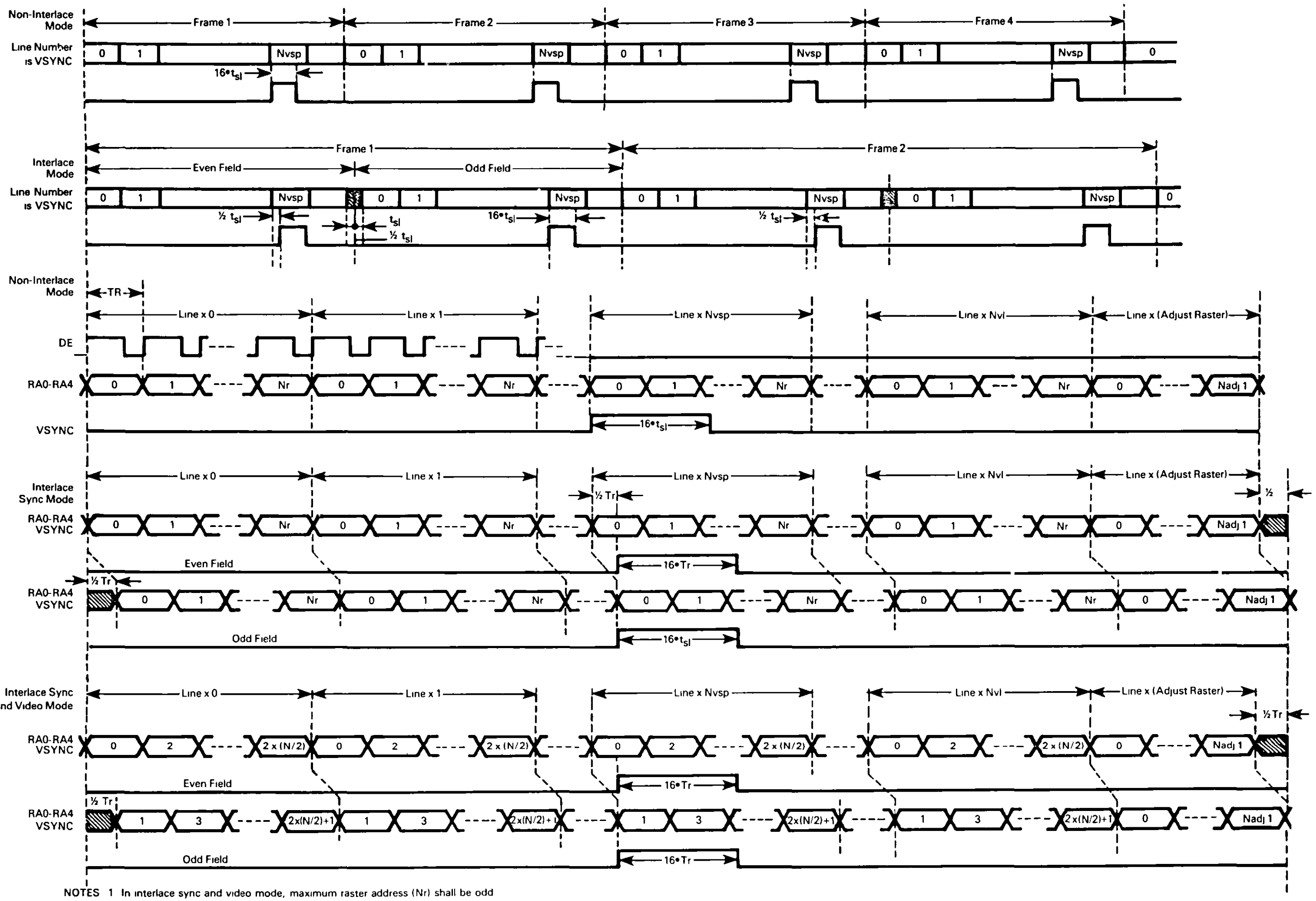
FIGURE 12 — CRTC HORIZONTAL TIMING


Horizontal Total (R0)
$t_{sl} = (N_{ht} + 1) \times t_c$
Horizontal Display (R1) $N_{hd} \times t_c$
Horizontal Retrace
t_c
CLK
MA0-MA13*
0
1
2
$N_{hd}-1$
N_{hd}
$N_{hsp}-1$
N_{hsp}
N_{ht}
Character #
0
1
2
$N_{hd}-1$
N_{hd}
$N_{hsp}-1$
N_{hsp}
N_{ht}
HSYNC
Horizontal Sync Position (R2)
Front Porch (Sync Delay)
HS Pulse Width (R3)
$N_{hsw} \times t_c$
Back Porch (Scan Delay)
Dispen

* Timing is shown for first displayed scan row only See chart in Figure 15 for other rows The initial MA is determined by the contents of start address register, R12/R13 Timing is shown for R12/R13 = 0

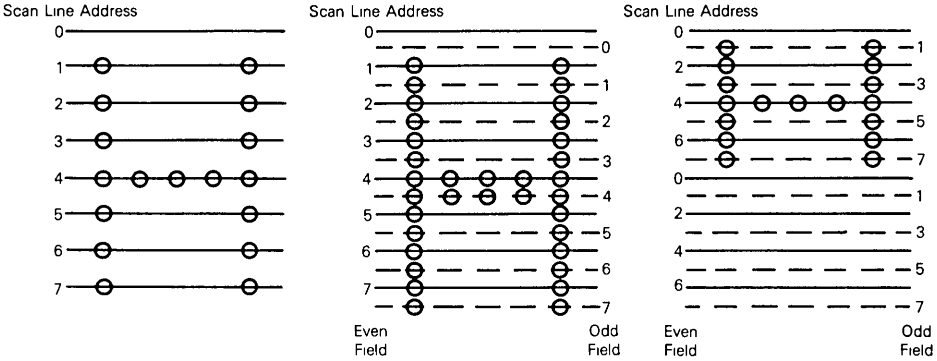
NOTE Timing values are described in Table 5

FIGURE 13 — CRTC VERTICAL TIMING


Non-Interlace Mode
Frame 1
Frame 2
Frame 3
Frame 4
Line Number is VSYNC
0
1
Nvsp
16•tsl
Interlace Mode
Even Field
Odd Field
½ tsl
tsl
Non-Interlace Mode
TR
Line x 0
Line x 1
Line x Nvsp
Line x Nvl
Line x (Adjust Raster)
DE
RA0-RA4
Nr
Nadj 1
VSYNC
Interlace Sync Mode
½ Tr
½
RA0-RA4 VSYNC
16•Tr
Interlace Sync and Video Mode
2
2 x (N/2)
3
2x(N/2)+1

NOTES 1 In interlace sync and video mode, maximum raster address (Nr) shall be odd
2 In interlace mode, Nht shall be odd

FIGURE 14 — INTERLACE CONTROL



(a) Normal Sync (b) Interlace Sync (c) Interlace Sync and Video

Maximum Scan Line Address Register (R9) — This 5-bit write-only register determines the number of scan lines per character row including the spacing, thus, controlling operation of the row address counter. The programmed value is a maximum address and is one less than the number of scan lines.

CURSOR CONTROL

Cursor Start Register (R10) and Cursor End Reigster (R11) — These registers allow a cursor of up to 32 scan lines in height to be placed on any scan line of the character block as shown in Figure 15 R10 is a 7-bit write-only register used to define the start scan line and the cursor blink rate. Bits 5 and 6 of the cursor start address register control the cursor operation as shown in Table 4. Non-display, display, and two blink modes (16 times or 32 times the field period) are available R11 is a 5-bit write-only register which defines the last scan line of the cursor.

TABLE 4 — CURSOR START REGISTER

Bit 6	Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate

Example of cursor display mode

When an external blink feature on characters is required, it may be necessary to perform cursor blink externally so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for a blinking cursor and externally inverting the video signal with an exclusive-OR gate.

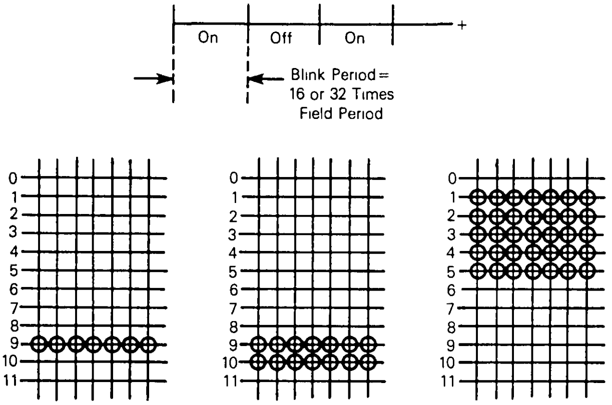
Cursor Register (R14-H, R15-L) — This 14-bit read/write register pair is programmed to position the cursor anywhere in the refresh RAM area; thus, allowing hardware paging and scrolling through memory without loss of the original cursor position. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register

OTHER REGISTERS

Start Address Register (R12-H, R13-L) — This 14-bit write-only register pair controls the first address output by the CRTC after vertical blanking. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. The start address register determines which portion of the refresh RAM is displayed on the CRT screen Hardware scrolling by character or page may be accomplished by modifying the contents of this register.

Light Pen Register (R16-H, R17-L) — This 14-bit read-only register pair captures the refresh address output by the CRTC on the positive edge of a pulse input to the LPSTB pin. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. Since the light pen pulse is asynchronous with respect to refresh address timing an internal synchronizer is designed into the CRTC. Due to delays (Figure 5) in this circuit, the value of R16 and R17 will need to be corrected in software Figure 16 shows an interrupt driven approach although a polling routine could be used

FIGURE 15 — CURSOR CONTROL

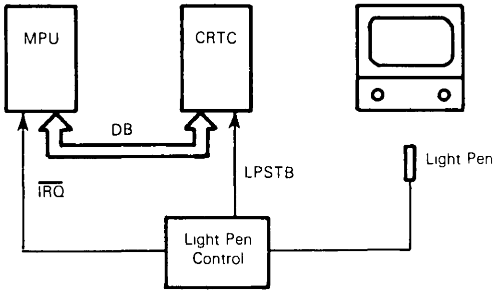

On
Off
On
+
Blink Period = 16 or 32 Times Field Period
0
1
2
3
4
5
6
7
8
9
10
11

Cursor Start Adr = 9
Cursor End Adr = 9

Cursor Start Adr = 9
Cursor End Adr = 10

Cursor Start Adr = 1
Cursor End Adr = 5

FIGURE 16 — INTERFACING OF LIGHT PEN


MPU
CRTC
DB
$\overline{IRQ}$
LPSTB
Light Pen
Light Pen Control

OPERATION OF THE CRTC

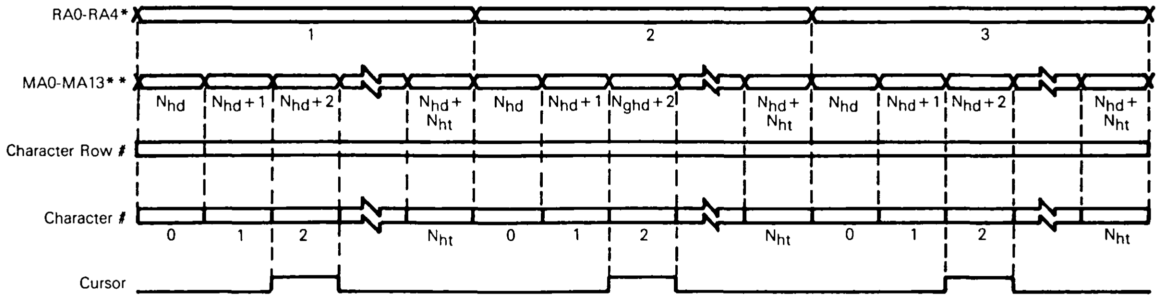
TIMING CHART OF THE CRT INTERFACE SIGNALS

Timing charts of CRT interface signals are illustrated in this section When values listed in Table 5 are programmed into CRTC control registers, the device provides the outputs as shown in the timing diagrams (Figures 12, 13, 17, and 18) The screen format is shown in Figure 11 which illustrates the relation between refresh memory address (MA0-MA13), raster address (RA0-RA4), and the position on the screen In this example, the start address is assumed to be zero

TABLE 5 — VALUES PROGRAMMED INTO CRTC REGISTERS

Reg. #	Register Name	Value	Programmed Value
R0	H Total	$N_{ht}+1$	N_{ht}
R1	H Displayed	N_{hd}	N_{hd}
R2	H Sync Position	N_{hsp}	N_{hsp}
R3	H Sync Width	N_{hsw}	N_{hsw}
R4	V Total	$N_{vt}+1$	N_{vt}
R5	V Scan Line Adjust	N_{adj}	N_{adj}
R6	V Displayed	N_{vd}	N_{vd}
R7	V Sync Position	N_{vsp}	N_{vsp}
R8	Interlace Mode		
R9	Max Scan Line Address	N_{sl}	N_{sl}

FIGURE 17 — CURSOR TIMING


RA0-RA4*
1
2
3
MA0-MA13**
Nhd
Nhd + 1
Nhd + 2
Nhd + Nht
Nhd
Nhd + 1
Nghd + 2
Nhd + Nht
Nhd
Nhd + 1
Nhd + 2
Nhd + Nht
Character Row #
Character #
0
1
2
Nht
0
1
2
Nht
0
1
2
Nht
Cursor

*Timing is shown for non-interlace and interlace sync modes
Example shown has cursor programmed as
Cursor Register = $N_{hd} + 2$
Cursor Start = 1
Cursor End = 3

**The initial MA is determined by the contents of start address register, R12/R13 Timing is shown for R12/R13 = 0

NOTE 1 Timing values are described in Table 5

FIGURE 18 — REFRESH MEMORY ADDRESSING (MA0-MA13) STAGE CHART

	Character Row	Scan Line	Horizontal Display				Horizontal Retrace (Non-Display)		
			Character						
Vertical Display	0	0	0	1	→	$N_{hd}-1$	N_{hd}	→	N_{ht}
		N_{sl}	0		→	$N_{hd}-1$	N_{hd}	→	N_{ht}
	1	0	N_{hd}	$N_{hd}+1$	→	$2XN_{hd}-1$	$2XN_{hd}$	→	$N_{hd}+N_{ht}$
		N_{sl}	N_{hd}	$N_{hd}+1$	→	$2XN_{hd}-1$	$2XN_{hd}$	→	$N_{hd}+N_{ht}$
	2	0	$2XN_{hd}$	$2XN_{hd}+1$	→	$3XN_{hd}-1$	$3XN_{hd}$	→	$2N_{hd}+N_{ht}$
		N_{sl}	$2XN_{hd}$	$2XN_{hd}+1$	→	$3XN_{hd}-1$	$3XN_{hd}$	→	$2N_{hd}+N_{ht}$
			↓	↓		↓	↓		↓
	$N_{vd}-1$	0	$(N_{vd}-1)\times N_{hd}$	$(N_{vd}-1)\times N_{hd}+1$	→	$N_{vd}\times N_{hd}+1$	$N_{vd}\times N_{hd}$	→	$(N_{vd}-1)\times N_{hd}+N_{ht}$
		N_{sl}	$(N_{vd}-1)\times N_{hd}$	$(N_{vd}-1)\times N_{hd}+1$	→	$N_{vd}\times N_{hd}-1$	$N_{vd}\times N_{hd}$	→	$(N_{vd}-1)\times N_{hd}+N_{ht}$
Vertical Retrace (Non-Display)	N_{vd}	0	$N_{vd}\times N_{hd}$	$N_{vd}\times N_{hd}+1$	→	$(N_{vd}+1)\times N_{hd}-1$	$(N_{vd}+1)\times N_{hd}$	→	$N_{vd}\times N_{hd}+N_{ht}$
		N_{sl}	$N_{vd}\times N_{hd}$	$N_{vd}\times N_{hd}+1$	→	$(N_{vd}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$	→	$N_{vd}+N_{hd}+N_{ht}$
	N_{vt}	0	$N_{vt}\times N_{hd}$	$N_{vt}\times N_{hd}+1$		$(N_{vt}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$		$N_{vt}\times N_{hd}+N_{ht}$
		N_{sl}	$N_{vt}\times N_{hd}$			$(N_{vt}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$		$N_{vt}\times N_{hd}+N_{ht}$
	$N_{vt}+1$	0	$(N_{vt}+1)\times N_{hd}$	$(N_{vt}+1)\times N_{hd}+1$		$(N_{vt}+2)\times N_{hd}-1$	$(N_{vt}+2)\times N_{hd}$		$(N_{v}+1)N_{hd}+N_{ht}$
		$N_{adj}-1$	$(N_{vt}+1)\times N_{hd}$	$(N_{vt}+1)\times N_{d}+1$		$(N_{vt}+2)\times N_{hd}-1$	$(N_{vt}+2)\times N_{hd}$		$(N_{vt}+1)N_{hd}+N_{ht}$

NOTE 1 The initial MA is determined by the contents of start address register, R12/R13 Timing is shown for R12/R13=0 Only non-interlace and interlace sync modes are shown

DETERMINING REGISTER CONTENTS

Some of the register contents are determined rather easily. They are:

Register	Name	Contents
R8	Interlace Mode Register	See Table 3
R10	Cursor Start	See Figure 15 and Table 4
R11	Cursor End	See Figure 15
R12	Start Address (H)	User programs first memory location to be displayed
R13	Start Address (L)	
R14	Cursor (H)	User programs desired cursor location
R15	Cursor (L)	
R16	Light Pen (H)	Can be loaded via light-pen strobe only
R17	Light Pen (L)	

The remaining register contents must be determined from some basic data related to the CRT monitor and from the user-desired display format. The CRTC reference sheet (see Figure 19) gives a set of formulas for calculating the register contents as well as other useful characteristics of the display. This type of data is summarized under basic parameters in Figures 20 and 21, most or all of this data must be supplied by the user before he can determine the contents for registers R0-R7 and R9. All variables B_1-B_{10} are equal to basic parameters 1 through 10.

FIGURE 19 — CRTC REFERENCE SHEET

Register Function		Intermediate Calculations			Register Calculations	
		Symbol	Description	Calculation	Register	Calculation
R0	Horizontal Total					
R1	Horizontal Displayed	f′	Dot frequency (1st approx.)	$\frac{B_5 \bullet (B_7+B_9)}{(1/B_1)-B_3}$	R0	$\frac{f'}{B_1 \bullet (B_7+B_9)}-1$
R2	Horizontal Sync Position					
R3	Horizontal Sync Width	t_c	Character Time	$\frac{1}{[(R0)+1] \bullet B_1}$	R1	B_5
R4	Vertical Total	f	Dot frequency	$\frac{B_7+B_9}{t_c}$		
R5	Vertical Total Adjust				R2	$(R1)+\frac{(R3)}{2}$
R6	Vertical Displayed	t_{sl}	Scan line time	$[(R0)+1] \bullet t_c$	R3	$\frac{(R0)-(R1)}{3}$
R7	Vertical Sync Position	n	Total # of scan lines	$\frac{1}{B_2 \bullet t_{sl}}$		
R8	Interlace Mode				R4	$N-1$
R9	Maximum Scan Line Address	N	Integer and	$\frac{n}{B_8+B_{10}} = N + \frac{R}{B_8+B_{10}}$	R5	R
R10	Cursor Start	R	Integer remainder			
R11	Cursor End				R6	B_6
R12	Start Address (H)	t_{cr}	Character row time	$(B_8+B_{10}) \bullet t_{sl}$	R7	$[(R4)+1] - \frac{16-(R5)}{B_8+B_{10}} \geq (R7) \geq (R6)$
R13	Start Address (L)	t_{hr}	Horizontal retrace time	$\leq \frac{[(R0)+1-B_5] \bullet (B_7+B_9)}{f}$		
R14	Cursor (H)				R9	$(B_8+B_{10})-1$
R15	Cursor (L)	t_{vr}	Vertical retrace time	$\leq \frac{B1}{B2} - B_6(B_8+B_{10}) \bullet t_{sl}$		
R16	Light Pen (H)					
R17	Light Pen (L)					

In Figures 20 and 21, worksheet example calculations are shown for 32 × 16 and 80 × 24 display formats respectively. The following items are keyed to the figures. Basic parameters ① through ⑩ have been provided, items ① through ④ are data about the CRT monitor and items ⑤ through ⑩ are data about the user's desired display.

① Calculate the approximate dot frequency. The user should verify that the bandwidth of his CRT monitor will accomodate this frequency

② Calculate R0. The resultant answer will usually be an integer plus a fraction. Assume the next high integer.

③ Fill in value for R1.

④ Calculate R3. Use the next highest integer In these examples the sync width was chosen to be one third of the horizontal blanking interval.

⑤ Calculate R2. Again, use the next highest integer

⑥ Calculate t_c, character tie. This is the time required for one scan line of one character block to be written.

⑦ Calculate the exact dot frequency.

⑧ Calculate t_{sl}, scan line time This is the time required for one scan line of one character row to be written including retrace time

⑨ Calculate n. This is the total number of scan lines for each frame Discard any fraction.'

⑩ Calculate N and R.

⑪ Calculate R4.

⑫ Fill in R5

⑬ Fill in R6

⑭ Calculate R7. If there is no possible value for R7, then the display demands for the CRT monitor exceed its capability A compromise adjustment must be made in basic parameter 6, 8, or 10

⑮ Calculate R9

⑯ Calculate t_{cr} This the time required for one character row to be written

⑰ Calculate t_{hr} $t_{hr} > B_3$

⑱ Calculate t_{vr} $t_{vr} > B_4$

In Figure 20, calculation Ⓑ verifies that the vertical period is 16 667 milliseconds or 60 hertz The expression used is

$$t_{cr} \times [(R4) + 1] + [t_{sl} \times (R5)] = V_P$$

Another check is calculation of horizontal sync pulse width R3 $t_c = PW_{HS}$ (typically approximately equals 4 microseconds)

For convenience, a blank worksheet is provided in Figure 22

FIGURE 20 — CRTC WORKSHEET EXAMPLE CALCULATION (32 × 16)

Basic Parameters (B1-B10)		Symbol		Intermediate Calculations	Value
1 Horizontal frequency	= 15750 ± 500	①	f'	$\dfrac{32 \times (5+2)}{\frac{1}{15750} - 11 \times 10^{-6}}$	= $4\ 27 \times 10^6$
2 Vertical frequency	= 60	⑥	t_c	$\dfrac{1}{39 \times 15750}$	= $1\ 63 \times 10^{-6}$
3 Minimum Horizontal retrace time	= 11×10^{-6}	⑦	f	$\dfrac{5+2}{1\ 63 \times 10-6}$	= $4\ 29 \times 10^6$
4. Minimum vertical retrace time	= 10^{-3}	⑧	t_{sl}	$39 \times 1\ 63 \times 10^{-6}$	= $63\ 6 \times 10^{-6}$
5 # of displayed characters per row	= 32	⑨	n	$\dfrac{1}{60 \times 63\ 6 \times 10^{-6}}$	= 262
6 # of displayed charactor rows	= 16	⑩	N		17
7 # of dots in character dot matrix row	= 5		R	$\dfrac{262}{7+8}$	= 7
8 # of scan lines in character • matrix column	= 7	⑯	t_{cr}	$(7+8) \times 63\ 6 \times 10^{-6}$	= 954×10^{-6}
9 Number of dots between horizontal adjacents	= 2	⑰	t_{hr}	$\dfrac{(38+1-32) \bullet (5+2)}{4\ 29 \times 10^6}$	= $11\ 42 \times 10^{-6}$
10 Number of scan lines between vertical adjacents	= 8	⑱	t_{vr}	$[\frac{15750}{60} - 16(7+8)] \times 63\ 6 \times 10^{-6}$	= $1\ 431 \times 10^{-3}$

Register Calculations			
Register		Decimal	Hex
② R0	$\dfrac{4\ 27 \times 10^6}{15{,}750 \times (5+2)} - 1 =$	38	26
③ R1	$B_5 = 32$	32	20
⑤ R2	$32 + \frac{3}{2}$	33	21
④ R3	$\frac{38-32}{3} =$	2	2
⑪ R4	$17 - 1 =$	16	10
⑫ R5	R = 7	7	7
⑬ R6	$B_6 = 16$	16	10
⑭ R7	Ⓐ	16	10
R8			
⑮ R9	$7 + 8 - 1 =$	14	0E
R10			
R11			
R12			
R13			
R14			
R15			

Ⓐ $16 + 1 - \dfrac{16-7}{7+8} \geq (R7) \geq 16$

$17 - 6 \geq (R7) \geq 16$

Ⓑ $$\begin{array}{l} 954 \times 10^{-6} \times 17 = 16\ 218 \times 10^{-3} \\ + \underline{63\ 6 \times 10^{-6} \times 7 = \ 445 \times 10^{-3}} \\ \quad 16\ 663 \text{ ms} \end{array} = \frac{1}{f}$$

$f \approx 60$ Hz

FIGURE 21 — CRTC WORKSHEET EXAMPLE CALCULATION (80 × 24)

Basic Parameters (B1-B10)			Symbol	Intermediate Calculations	Value
1 Horizontal frequency	=	18,600	① f′	$\frac{80\bullet(7+2)}{\frac{1}{18600}-11\times10^{-6}}$	$16\bullet836\times10^{6}$
2 Vertical frequency	=	60	⑥ t_c	$\frac{1}{(100+1)\bullet18600}$	$532\ 31\times10^{-9}$
3 Minimum Horizontal retrace time	=	11×10^{-6}	⑦ f	$\frac{7+2}{532\ 31\times10^{-9}}$	$16\ 907\times10^{6}$
4 Minimum vertical retrace time	=	1×10^{-3}	⑧ t_{sl}	$(100+1)(532\ 31\times10^{-9})$	$53\ 76\times10^{-6}$
5 # of displayed characters per row	=	80	⑨ n	$\frac{1}{(60)(53\ 76\times10^{-6})}$	310
6 # of displayed charactor rows	=	24	⑩ N		28
7 # of dots in character dot matrix row	=	7	R	$\frac{310}{11}$	2
8 # of scan lines in charactor • matrix column	=	9	⑯ t_{cr}	$(9+2)(53\ 76\times10^{-6})$	$591\ 39\times10^{-6}$
9 Number of dots between horizontal adjacents	=	2	⑰ t_{hr}	$\le \frac{(101-80)(7+2)}{16\ 907\times10^{6}}$	$11\ 17\times10^{-6}$
10 Number of scan lines between vertical adjacents	=	2	⑱ t_{vr}	$\le \left[\frac{18600}{60}-24(11)\right]53\ 76\times10^{-6}$	$2\ 47\times10^{-3}$

Register	Register Calculations	Decimal	Hex
② R0	$\frac{16\ 836\times10^{6}}{(18,600)(9)}-1$	100	64
③ R1	B5 = 80	80	50
⑤ R2	$80+\frac{7}{2}$	84	54
④ R3	$\frac{R0-R1}{3}$	7	07
⑪ R4	28 − 1	27	1B
⑫ R5	R = 2	02	02
⑬ R6	B6 = 24	24	18
⑭ R7	Ⓐ	25	19
R8		0	0
⑮ R9	(9 + 2) − 1	10	0A
R10		00	00
R11		11	0B
R12			00
R13		128	80
R14		128	00
R15			80

Ⓐ $(27+1)-\frac{(16-2)}{11} \ge R7 \ge 24$

$26\ 72 \ge R7 \ge 24$

Ⓑ $B2 = 1/[(t_{cr})(R4+1)+(t_{sl})(R5)]$

$= 1/[(591\ 39\times10^{6})(28)+(53\ 76\times10^{-6})(2)]$

$= 1/16\ 667\times10^{-3}$

$= 60$

FIGURE 22 — CRTC WORKSHEET

Basic Parameters			Symbol	Intermediate Calculations Value
1 Horizontal frequency	=	________	f'	________
2 Vertical frequency	=	________	t_c	________
3 Minimum Horizontal retrace time	=	________	f	________
4 Minimum vertical retrace time	=	________	t_{sl}	________
5 # of displayed characters per row	=	________	n	________
6. # of displayed charactor rows	=	________	N	________
7 # of dots in character dot matrix row	=	________	R	________
8 # of scan lines in char-actor • matrix column	=	________	t_{cr}	________
9 Number of dots between horizontal adjacents	=	________	t_{hr}	________
10 Number of scan lines between vertical adjacents	=	________	t_{vr}	________

Register Calculations Register	Decimal	Hex
R0	______	______
R1	______	______
R2	______	______
R3	______	______
R4	______	______
R5	______	______
R6	______	______
R7	______	______
R8	______	______
R9	______	______
R10	______	______
R11	______	______
R12	______	______
R13	______	______
R14	______	______
R15	______	______
R16	______	______
R17	______	______
R18	______	______
R19	______	______

CRTC INITIALIZATION

Register R0-R15 must be initialized after the system is powered up. The processor will normally load the CRTC register file from a firmware table. The program required to initialize the CRTC for a 80 × 24 format (example calculation #2) is shown in Figure 23

The CRTC registers will have an initial value at power up When using a direct drive monitor (sans horizontal oscillator) these initial values may result in out-of-tolerance operation CRTC programming should be done immediately after power up especially in this type of system

ADDITIONAL CRTC APPLICATIONS

The foremost system function which may be performed by the CRTC controller is the refreshing of dynamic RAM This is quite simple as the refresh addresses continually run

Note that the LPSTB input may be used to support additional system functions other than a light pen A digital-to-analog converter (DAC) and comparator could be configured to use the refresh addresses as a reference to a DAC composed of a resistive adder network connected to a comparator The output of the comparator would generate the LPSTB input signifying a match between the refresh address analog level and the unknown voltage

The light-pen strobe input could also be used as a character strobe to allow the CRTC refresh addresses to decode a keyboard matrix Debouncing would need to be done in software

Both the VS and HS outputs may be used as a real-time clock Once programmed, the CRTC will provide a stable reference frequency

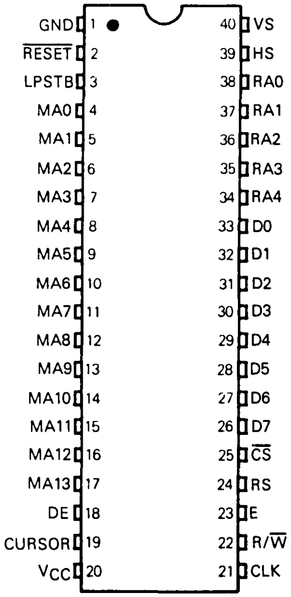
FIGURE 23 — MC6800 PROGRAM FOR CRTC INITIALIZATION

```
PAGE 001  CRTCINIT.SA:0   MC6845 CRTC Initialization Program

00001                          NAM    MC6845
00002                          TTL    / MC6845-1 CRTC initialization program
00003                          OPT    G,S,LLE=85 print FCB's, FDB's & XREF table
00004                   ********************************************************
00005                   * Assign CRTC addresses
00006                   *
00007          9000   A CRTCAD EQU    $9000     Address Register
00008          9001   A CRTCRG EQU    CRTCAD+1  Data Register
00009                   ********************************************************
00010                   * Initialization program
00011                   *
00012A 0000                    ORG    0         a place to start
00013A 0000 5F                 CLRB             clear counter
00014A 0001 CE 1020   A        LDX    #CRTTAB   table pointer
00015A 0004 F7 9000   A CRTC1  STAB   CRTCAD    load address register
00016A 0007 A6 00     A        LDAA   0,X       get register value from table
00017A 0009 B7 9001   A        STAA   CRTCRG    program register
00018A 000C 08                 INX              increment counters
00019A 000D 5C                 INCB
00020A 000E C1 10     A        CMPB   $10       finished?
00021A 0010 26 F2 0004         BNE    CRTC1     no: take branch
00022A 0012 3F                 SWI              yes: call monitor
00023                   ********************************************************
00024                   * CRTC register initialization table
00025                   * 80 x 24 non-interlaced format
00026A 1020                    ORG    $1020     start of table
00027A 1020    65     A CRTTAB FCB    $64,$50   R0, R1 - H total & H displayed
     A 1021    50     A
00028A 1022    56     A        FCB    $54,$07   R2, R3 - HS pos. & HS width
     A 1023    09     A
00029A 1024    18     A        FCB    $1B,$02   R4, R5 - V total & V total adj.
     A 1025    0A     A
00030A 1026    18     A        FCB    $18,$19   R6, R7 - V displayed $ VS pos.
     A 1027    18     A
00031A 1028    00     A        FCB    $00,$0A   R8, R9 - Interlace & Max scan line
     A 1029    0B     A
00032A 102A    00     A        FCB    $00,$0B   R10,R11- Cursor start & end
     A 102B    0B     A
00033A 102C    0080   A        FDB    $0080     R12,R13- Start Address
00034A 102E    0080   A        FDB    $0080     R14,R15- Cursor Address
00035                          END
TOTAL ERRORS 00000--00000

CRTC1   0004   CRTCAD 9000   CRTCRG 9001   CRTTAB 1020
```

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip S Suffix	1.0	0°C to 70°C	MC6845S
	1.0	−40°C to +85°C	MC6845CS
	1.5	0°C to 70°C	MC68A45S
	1.5	−40°C to +85°C	MC68A45CS
	2.0	0°C to 70°C	MC68B45S
Plastic P Suffix	1.0	0°C to 70°C	MC6845P
	1.0	−40°C to +85°C	MC6845CP
	1.5	0°C to 70°C	MC68A45P
	1.5	−40°C to +85°C	MC68A45CP
	2.0	0°C to 70°C	MC68B45P

PIN ASSIGNMENT


GND 1
RESET 2
LPSTB 3
MA0 4
MA1 5
MA2 6
MA3 7
MA4 8
MA5 9
MA6 10
MA7 11
MA8 12
MA9 13
MA10 14
MA11 15
MA12 16
MA13 17
DE 18
CURSOR 19
VCC 20
40 VS
39 HS
38 RA0
37 RA1
36 RA2
35 RA3
34 RA4
33 D0
32 D1
31 D2
30 D3
29 D4
28 D5
27 D6
26 D7
25 CS
24 RS
23 E
22 R/W
21 CLK

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC6850

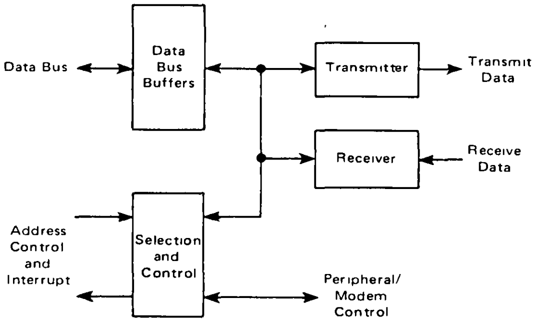
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional −1, −16, and −64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6850, MC68A50, MC68B50 MC6850C, MC68A50C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC})

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 120 65	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts – Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, CS0, CS1, $\overline{CS2}$, Enable RS, Rx D, Rx C, $\overline{CTS}$, $\overline{DCD}$	I_{in}	–	1.0	2.5	µA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	µA
Output High Voltage (I_{Load} = −205 µA, Enable Pulse Width < 25 µs) (I_{Load} = −100 µA, Enable Pulse Width < 25 µs)	 D0-D7 Tx Data, $\overline{RTS}$	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$	 – –	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 µs)		V_{OL}	–	–	$V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	µA
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	300	525*	mW
Internal Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 E, Tx CLK, Rx CLK, R/$\overline{W}$, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$	C_{in}	 – –	 10 7.0	 12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	RTS, Tx Data $\overline{IRQ}$	C_{out}	– –	– –	10 5.0	pF

*For temperatures less than T_A = 0°C, P_{INT} maximum will increase.

SERIAL DATA TIMING CHARACTERISTICS

Characteristic		Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low (See Figure 1)	÷16, ÷64 Modes ÷1 Mode	PW_{CL}	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Pulse Width, High (See Figure 2)	÷16, ÷64 Modes ÷1 Mode	PW_{CH}	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Frequency	÷16, ÷64 Modes ÷1 Mode	f_C	– –	0.8 500	– –	1.0 750	– –	1.5 1000	MHz kHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)		t_{TDD}	–	600	–	540	–	460	ns
Receive Data Setup Time (See Figure 4)	÷1 Mode	t_{RDS}	250	–	100	–	30	–	ns
Receive Data Hold Time (See Figure 5)	÷1 Mode	t_{RDH}	250	–	100	–	30	–	ns
Interrupt Request Release Time (See Figure 6)		t_{IR}	–	1.2	–	0.9	–	0.7	µs
Request-to-Send Delay Time (See Figure 6)		t_{RTS}	–	560	–	480	–	400	ns
Input Rise and Fall Times (or 10% of the pulse width if smaller)		t_r, t_f	–	1.0	–	0.5	–	0.25	µs

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

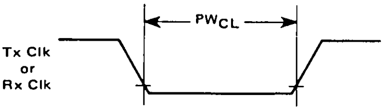


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

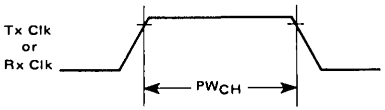


FIGURE 3 — TRANSMIT DATA OUTPUT DELAY

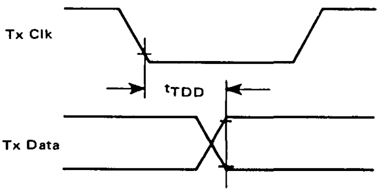


FIGURE 4 — RECEIVE DATA SETUP TIME (÷1 Mode)

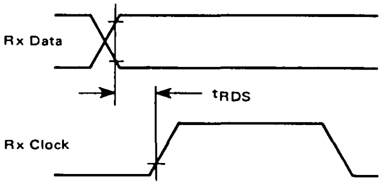


FIGURE 5 — RECEIVE DATA HOLD TIME (÷1 Mode)

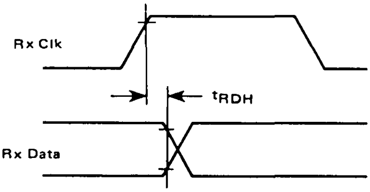
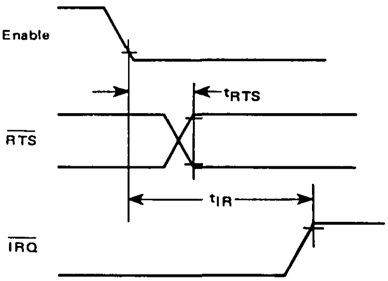


FIGURE 6 — REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES



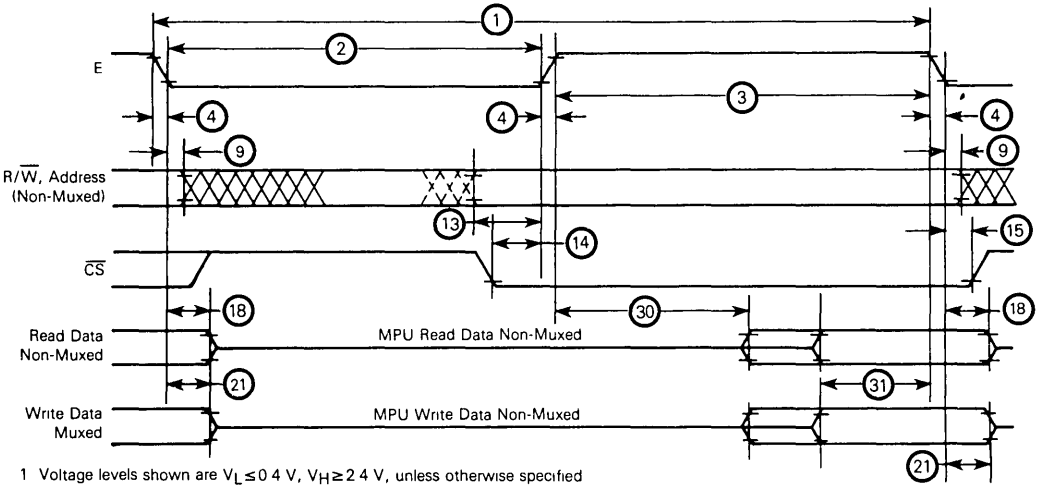
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

BUS TIMING CHARACTERISTICS (See Notes 1 and 2 and Figure 7)

Ident. Number	Characteristic	Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance)

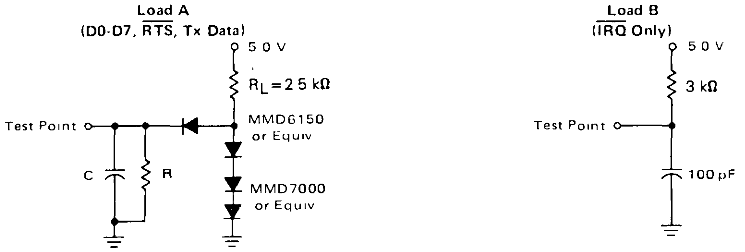
FIGURE 7 — BUS TIMING CHARACTERISTICS



E
①
②
③
④
④
④
⑨
⑨
R/$\overline{W}$, Address (Non-Muxed)
⑬
⑭
⑮
$\overline{CS}$
⑱
⑳
⑱
Read Data Non-Muxed
MPU Read Data Non-Muxed
㉑
㉛
Write Data Muxed
MPU Write Data Non-Muxed
㉑

1 Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified
2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FIGURE 8 — BUS TIMING TEST LOADS

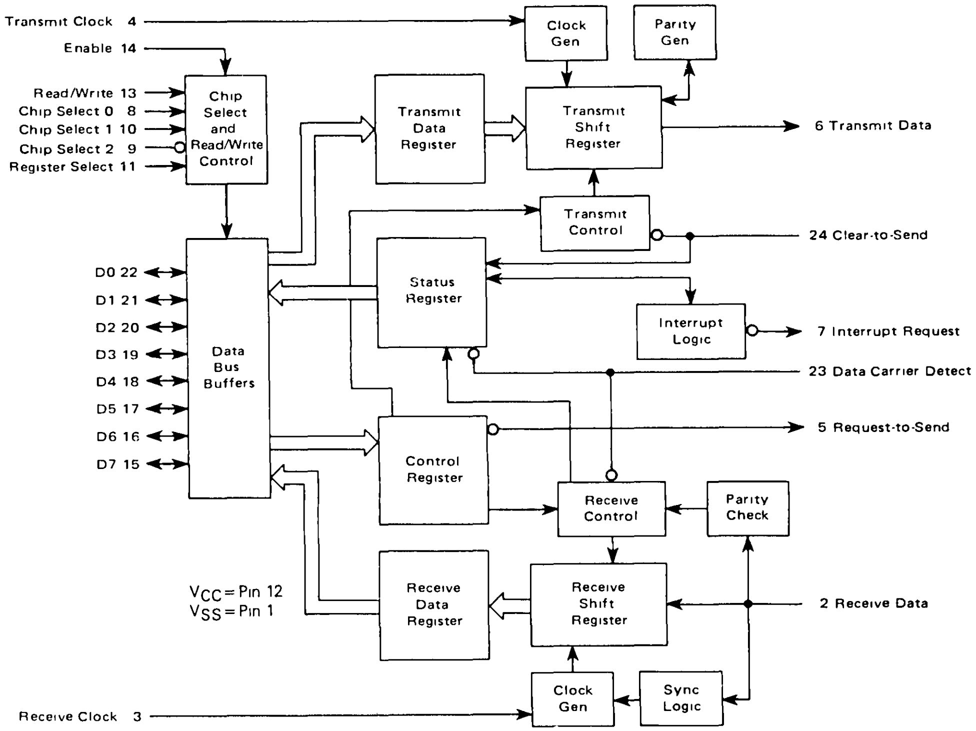


Load A
(D0-D7, $\overline{RTS}$, Tx Data)
5.0 V
$R_L = 2.5$ kΩ
MMD6150 or Equiv
Test Point
C
R
MMD7000 or Equiv
Load B
($\overline{IRQ}$ Only)
5.0 V
3 kΩ
Test Point
100 pF

C = 130 pF for D0-D7
= 30 pF for $\overline{RTS}$ and Tx Data

R = 11.7 kΩ for D0-D7
= 24 kΩ for $\overline{RTS}$ and Tx Data

FIGURE 9 — EXPANDED BLOCK DIAGRAM


Transmit Clock 4
Enable 14
Read/Write 13
Chip Select 0 8
Chip Select 1 10
Chip Select 2 9
Register Select 11
Chip Select and Read/Write Control
D0 22
D1 21
D2 20
D3 19
D4 18
D5 17
D6 16
D7 15
Data Bus Buffers
Clock Gen
Parity Gen
Transmit Data Register
Transmit Shift Register
6 Transmit Data
Transmit Control
24 Clear-to-Send
Status Register
Interrupt Logic
7 Interrupt Request
23 Data Carrier Detect
5 Request-to-Send
Control Register
Receive Control
Parity Check
V_{CC} = Pin 12
V_{SS} = Pin 1
Receive Data Register
Receive Shift Register
2 Receive Data
Clock Gen
Sync Logic
Receive Clock 3

DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations Internally, there are four registers two read-only and two write-only registers The read-only registers are Status and Receive Data, the write-only registers are Control and Transmit Data The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines

MASTER RESET

The master reset (CR0, CR1) must be set immediately after power-up to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. During the first master reset, the $\overline{\mathrm{IRQ}}$ and $\overline{\mathrm{RTS}}$ outputs are held at level 1. On all other master resets, the $\overline{\mathrm{RTS}}$ output can be programmed high or low with the $\overline{\mathrm{IRQ}}$ output held high Control bits CR5 and CR6 should also be programmed to define the state of $\overline{\mathrm{RTS}}$ whenever master reset is utilized After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits, and parity (even, odd, or none)

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status If the register is empty, another character can be loaded for transmission even through the first character is in the process of being transmitted (because of double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of 8 or 32 low samples on the receive line in the divide-by-16 and 64 modes respectively. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for a 7-bit word (7 bits plus parity), the receiver strips the parity bit (D7 = 0) so that data alone is transferred to the MPU This feature reduces MPU programming. The Status Register can continue to be read to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the M6800 MPU with an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (D0-D7) – The bidirectional data lines (D0-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) – The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 $\phi 2$ Clock or MC6809 E clock.

Read/Write (R/$\overline{\text{W}}$) – The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, $\overline{\text{CS2}}$) – These three high-impedance TTL-compatible input lines are used to address the ACIA The ACIA is selected when CS0 and CS1 are high and $\overline{\text{CS2}}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable Signal, Read/Write, and Register Select

Register Select (RS) – The Register Select line is a high-impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request ($\overline{\text{IRQ}}$) – Interrupt Request is a TTL-compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU The $\overline{\text{IRQ}}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The IRQ status bit, when high, indicates the $\overline{\text{IRQ}}$ output is in the active state

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5•$\overline{\text{CR6}}$), and the Transmit Data Register Empty (TDRE) status bit is high The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send ($\overline{\text{CTS}}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmit Data Register The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of $\overline{\text{CTS}}$ which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect ($\overline{\text{DCD}}$) has gone high An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA Interrupts caused by Overrun or loss of $\overline{\text{DCD}}$ are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16, or 64 times the data rate may be selected.

Transmit Clock (Tx CLK) – The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK) – The Receive Clock input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally) The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) — The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used.

Transmit Data (Tx Data) — The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send ($\overline{CTS}$) — This high-impedance TTL-compatible input provides automatic control of the transmitting end of a communications link via the modem Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send ($\overline{RTS}$) — The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The RTS output corresponds to the state of the Control Register bits CR5 and CR6 When CR6=0 or both CR5 and CR6=1, the $\overline{RTS}$ output is low (the active state) This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect ($\overline{DCD}$) — This high-impedance TTL-compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output The $\overline{DCD}$ input inhibits and initializes the receiver section of the ACIA when high. A low-to-high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set. The Rx CLK must be running for proper $\overline{DCD}$ operation.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed with RS high and R/$\overline{W}$ low. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 1-bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/$\overline{W}$ high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

TABLE 1 — DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	$RS \bullet \overline{R/\overline{W}}$ Transmit Data Register	$RS \bullet R/\overline{W}$ Receive Data Register	$\overline{RS} \bullet \overline{R/\overline{W}}$ Control Register	$\overline{RS} \bullet R/\overline{W}$ Status Register
	(Write Only)	(Read Only)	(Write Only)	(Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send ($\overline{CTS}$)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request (IRQ)

* Leading bit = LSB = Bit 0

** Data bit will be zero in 7 bit plus parity modes

*** Data bit is "don't care" in 7 bit plus parity modes

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/$\overline{\mathrm{W}}$ are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1) – The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{\mathrm{CTS}}$ and $\overline{\mathrm{DCD}}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	÷1
0	1	÷16
1	0	÷64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) – The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows.

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send ($\overline{\mathrm{RTS}}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{\mathrm{RTS}}$ = low, Transmitting Interrupt Disabled
0	1	$\overline{\mathrm{RTS}}$ = low, Transmitting Interrupt Enabled.
1	0	$\overline{\mathrm{RTS}}$ = high, Transmitting Interrupt Disabled.
1	1	$\overline{\mathrm{RTS}}$ = low, Transmits a Break level on the Transmit Data Output Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7) – The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low-to-high transition on the Data Carrier Detect ($\overline{\mathrm{DCD}}$) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/$\overline{\mathrm{W}}$ is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 – Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current Data Carrier Detect being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 – The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command

Data Carrier Detect (DCD), Bit 2 – The Data Carrier Detect bit will be high when the $\overline{\mathrm{DCD}}$ input from a modem has gone high to indicate that a carrier is not present This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{\mathrm{DCD}}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{\mathrm{DCD}}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the DCD status bit remains high and will follow the $\overline{\mathrm{DCD}}$ input

Clear-to-Send ($\overline{\mathrm{CTS}}$), Bit 3 – The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem A low $\overline{\mathrm{CTS}}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high Master reset does not affect the Clear-to-Send status bit

Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the first stop bit This error indicates a synchronization error, faulty transmission, or a break condition The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

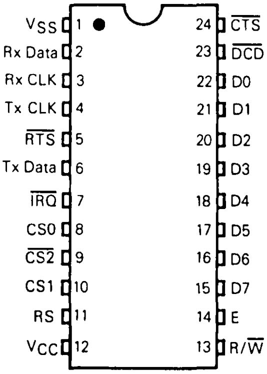
Parity Error (PE), Bit 6 — The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver partiy check results are inhibited.

Interrupt Request (IRQ), Bit 7 — The IRQ bit indicates the state of the $\overline{\text{IRQ}}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{\text{IRQ}}$ output is low the IRQ bit will be high to indicate the interrupt or service request status. $\overline{\text{IRQ}}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip S Suffix	1.0	0°C to 70°C	MC6850S
	1.0	−40°C to 85°C	MC6850CS
	1.5	0°C to 70°C	MC68A50S
	1.5	−40°C to 85°C	MC68A50CS
	2.0	0°C to 70°C	MC68B50S
Plastic P Suffix	1.0	0°C to 70°C	MC6850P
	1.0	−40°C to 85°C	MC6850CP
	1.5	0°C to 70°C	MC68A50P
	1.5	−40°C to 85°C	MC68A50CP
	2.0	0°C to 70°C	MC68B50P

PIN ASSIGNMENTS



MC6852

Synchronous Serial Data Adapter (SSDA)

The MC6852 Synchronous Serial Data Adaper provides a bidirectional serial interface for synchronous data information interchange. It contains interface logic for simultaneously transmitting a receiving standard synchronous communications characters in bus organized systems such as the M6800 Microprocessor systems.

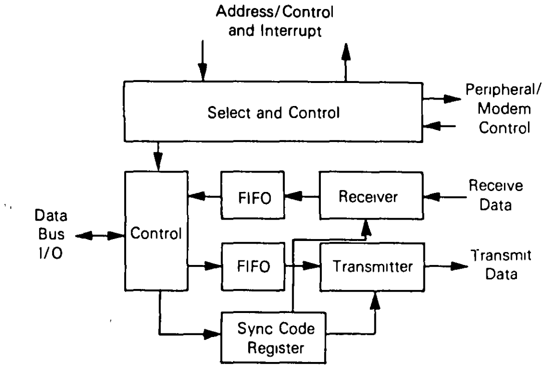
The bus interface of the MC6852 includes select, enable, read/write, interrupt, and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the synchronous data interface with synchronization, fill character insertion/deletion, and error checking. The functional configuration of the SSDA is programmed via the data bus during system initialization. Programmable control registers provide control for variable word lengths, transmit control, receive control, synchronization control, and interrupt control. Status, timing and control lines provide peripheral or modem control.

Typical applications include floppy disk controllers, cassette or cartridge tape controllers, data communications terminals, and numerical control systems.

- Programmable Interrupts from Transmitter, Receiver, and Error Detection Logic
- Character Synchronization on One- or Two-Sync Codes
- External Synchronization Available for Parallel-Serial Operation
- Programmable Sync Code Register
- Up to 1.5 MHz Transmission
- Peripheral/Modem Control Functions
- Three Bytes of FIFO Buffering on Both Transmit and Receive
- 7-, 8-, or 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun, and Underflow Status

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SYNCHRONOUS SERIAL DATA ADAPTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6852, MC68A52, MC68B52 MC6852C, MC68A52C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic Package		120	
Cerdip Package		65	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advsied that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \qquad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	–	V
Input Low Voltage		V_{IL}	–	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	Tx CLK, Rx CLK, Rx Data, Enable, $\overline{\text{RESET}}$, RS, R/$\overline{\text{W}}$, $\overline{\text{CS}}$, $\overline{\text{DCD}}$, $\overline{\text{CTS}}$	I_{in}	–	1.0	2.5	μA
Hi-Z (Off-State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.25 V)	D0-D7	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = –205 μA, Enable Pulse Width < 25 μs)	D0-D7	V_{OH}	V_{SS} + 2.4	–	–	V
(I_{Load} = –100 μA, Enable Pulse Width < 25 μs)	TX Data, $\overline{\text{DTR}}$, TUF		V_{SS} + 2.4	–	–	
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 μs)		V_{OL}	–	–	V_{SS} + 0.4	V
Output Leakage Current (Off-State) (V_{OH} = 2.4 V)	$\overline{\text{IRQ}}$	I_{OZ}	–	1.0	10	μA
Internal Power Dissipation (Measured at T_A = 0°C)*		P_{INT}	–	500	630*	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Other Inputs		–	–	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	Tx Data, SM/$\overline{\text{DTR}}$, TUF	C_{out}	–	–	10	pF
	$\overline{\text{IRQ}}$				5.0	

*For temperatures below 0°C, the maximum value of P_{INT} will increase

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	MC6852		MC68A52		MC68B52		Unit
		Min	Max	Min	Max	Min	Max	
Serial Clock Pulse Width, Low (Figure 1)	PW_{CL}	700	–	400	–	280	–	ns
Serial Clock Pulse Width, High (Figure 2)	PW_{CH}	700	–	400	–	280	–	ns
Serial Clock Frequency (Rx CLK, Tx CLK)	f_C	–	600	–	1000	–	1500	kHz
Receive Data Setup Time (Figure 3, 7)	t_{RDSU}	350	–	200	–	160	–	ns
Receive Data Hold Time (Figure 3)	t_{RDH}	350	–	200	–	160	–	ns
Sync Match Delay Time (Figure 3)	t_{SM}	–	1.0	–	0.666	–	0.500	μs
Clock-to-Data Delay for Transmitter (Figure 4)	T_{DD}	–	1.0	–	0.666	–	0.500	μs
Transmitter Underflow (Figures 4, 6)	t_{TUF}	–	1.0	–	0.666	–	0.500	μs
$\overline{\text{DTR}}$ Delay Time (Figure 5)	t_{DTR}	–	1.0	–	0.666	–	0.500	μs
Interrupt Request Release Time (Figure 5)	t_{IR}	–	1.6	–	1.1	–	0.850	μs
$\overline{\text{RESET}}$ Pulse Width	t_{RESET}	1.0	–	0.666	–	0.500	–	μs
$\overline{\text{CTS}}$ Setup Time (Figure 6)	t_{CTS}	200	–	150	–	120	–	ns
$\overline{\text{DCD}}$ Setup Time (Figure 7)	t_{DCD}	500	–	350	–	250	–	ns
Input Rise and Fall Times (Except Enable)	t_r, t_f	–	1.0*	–	1.0*	–	1.0*	μs

*1.0 μs or 10% of the pulse width, whichever is smaller

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

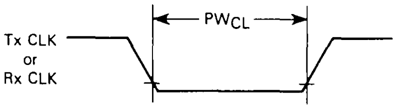
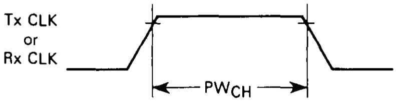


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE



Note Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

FIGURE 3 — RECEIVE DATA SETUP AND HOLD TIMES AND SYNC MATCH DELAY TIME

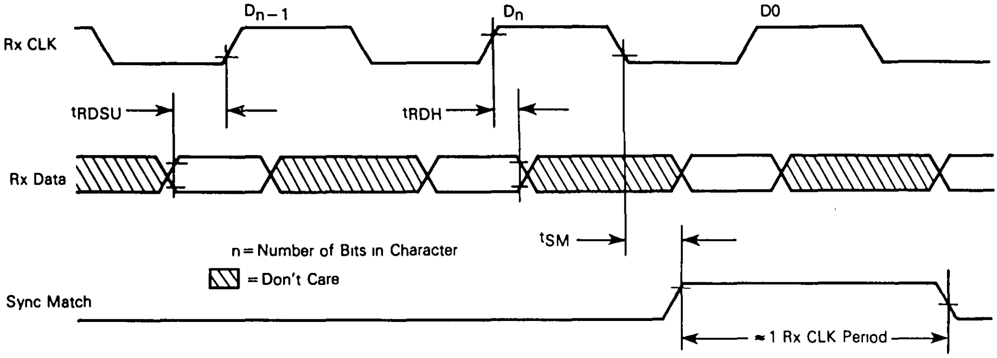
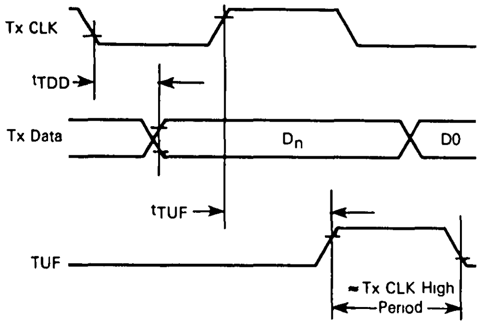

Rx CLK
D_{n-1}
D_n
D0
t_{RDSU}
t_{RDH}
Rx Data
t_{SM}
n = Number of Bits in Character
▧ = Don't Care
Sync Match
≈ 1 Rx CLK Period

FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND TRANSMITTER UNDERFLOW DELAY TIME


Tx CLK
t_{TDD}
Tx Data
D_n
D0
t_{TUF}
TUF
≈ Tx CLK High Period

n = Number of bits in character

FIGURE 5 — DATA TERMINAL READY AND INTERRUPT REQUEST RELEASE TIMES

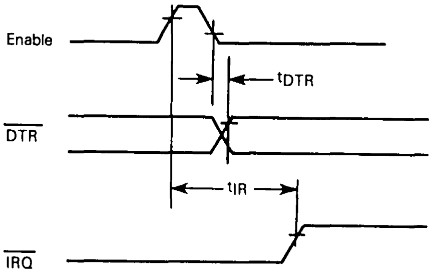

Enable
t_{DTR}
$\overline{DTR}$
t_{IR}
$\overline{IRQ}$

FIGURE 6 — CLEAR-TO-SEND SETUP TIME

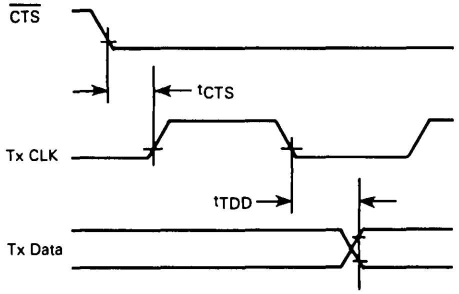
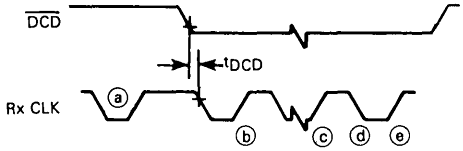

$\overline{CTS}$
t_{CTS}
Tx CLK
t_{TDD}
Tx Data

FIGURE 7 — DATA CARRIER DETECT SETUP TIME


$\overline{DCD}$
t_{DCD}
Rx CLK
ⓐ
ⓑ
ⓒ
ⓓ
ⓔ

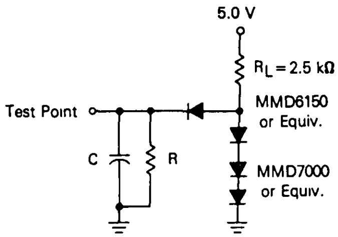
Notes.
- a Must occur before $\overline{DCD}$ goes low.
- b First data bit placed in Rx shift register
- c. Last data bit of byte placed in Rx shift register.
- d. Rx data byte transferred from shift register to Rx FIFO.
- e Clock edge required for generation of $\overline{IRQ}$ by RDA status

Note: Refer to Figure 3 for the Rx data setup and hold times

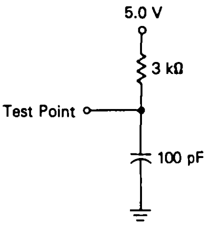
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2 0 volts, unless otherwise noted.

BUS TIMING TEST LOADS

Load A
(D0-D7, $\overline{DTR}$, Tx Data, TUF)


5.0 V
$R_L = 2.5\ k\Omega$
MMD6150
or Equiv.
Test Point
C
R
MMD7000
or Equiv.

Load B
($\overline{IRQ}$ Only)


5.0 V
3 kΩ
Test Point
100 pF

C = 130 pF for D0-D7
= 30 pF for $\overline{DTR}$, Tx Data, and TUF

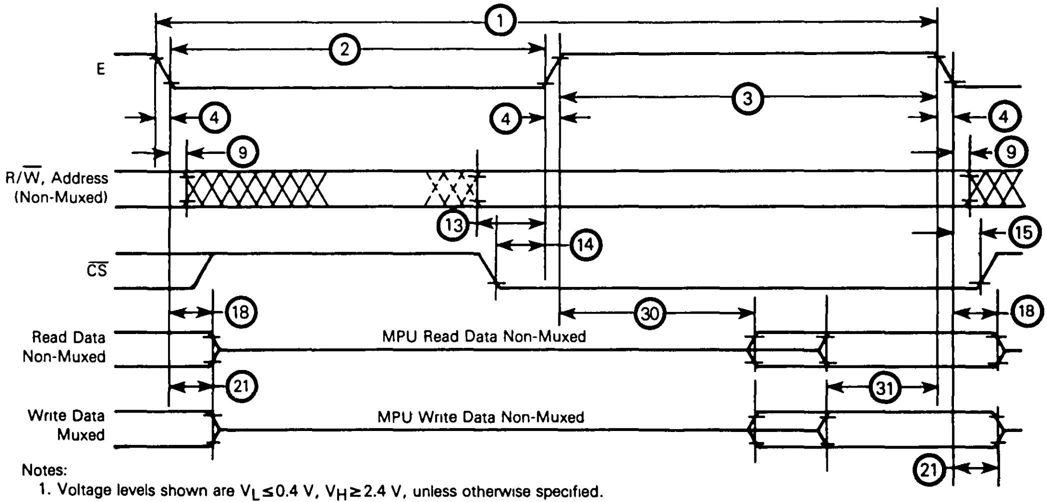
R = 11.7 kΩ for D0-D7
= 24 kΩ for $\overline{DTR}$, Tx Data, and TUF

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Indent Number	Characteristic	Symbol	MC6852		MC68A52		MC68B52		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	30	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

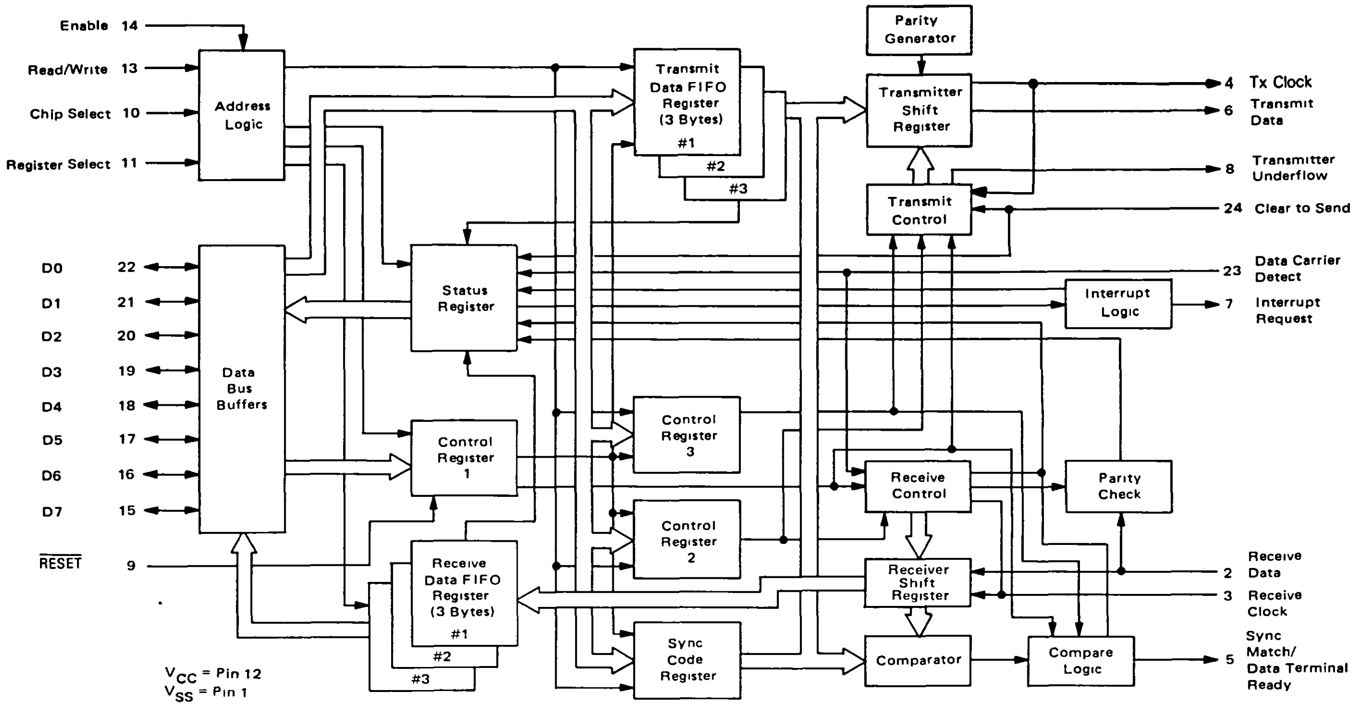
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 8 — BUS TIMING CHARACTERISTICS
(READ/WRITE INFORMATION)


①
②
③
④
⑨
⑬
⑭
⑮
⑱
㉑
㉚
㉛
E
R/$\overline{W}$, Address
(Non-Muxed)
$\overline{CS}$
Read Data
Non-Muxed
MPU Read Data Non-Muxed
Write Data
Muxed
MPU Write Data Non-Muxed

Notes:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

EXPANDED BLOCK DIAGRAM



Enable 14
Read/Write 13
Chip Select 10
Register Select 11
Address Logic
D0 22
D1 21
D2 20
D3 19
D4 18
D5 17
D6 16
D7 15
Data Bus Buffers
RESET 9
V_{CC} = Pin 12
V_{SS} = Pin 1
Status Register
Control Register 1
Receive Data FIFO Register (3 Bytes)
#1
#2
#3
Transmit Data FIFO Register (3 Bytes)
#1
#2
#3
Control Register 3
Control Register 2
Sync Code Register
Parity Generator
Transmitter Shift Register
Transmit Control
Interrupt Logic
Receive Control
Parity Check
Receiver Shift Register
Comparator
Compare Logic
4 Tx Clock
6 Transmit Data
8 Transmitter Underflow
24 Clear to Send
23 Data Carrier Detect
7 Interrupt Request
2 Receive Data
3 Receive Clock
5 Sync Match/Data Terminal Ready

DEVICE OPERATION

At the bus interface, the SSDA appears as two addressable memory locations. Internally, there are seven registers: two read-only and five write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control 1, Control 2, Control 3, Sync Code and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and four peripheral/modem control lines.

Data to be transmitted is transferred directly into the 3-byte Transmit Data First-In First-Out (FIFO) Register from the data bus. Availability of the input to the FIFO is indicated by the TDRA bit in the Status Register; once data is entered, it moves through the FIFO to the last empty location. Data at the output of the FIFO is automatically transferred from the FIFO to the Transmitter Shift Register as the shift register becomes available to transmit the next character. If data is not available from the FIFO (underflow condition), the Transmitter Shift Register is automatically loaded with either a sync code or an all "1's" character. The transmit seciton may be programmed to append even, odd, or no parity to the transmitted word. An external control line (Clear-to-Send) is provided to inhibit the transmitter without clearing the FIFO.

Serial data is accumulated in the receiver based on the synchronization mode selected. In the external sync mode, used for parallel-serial operation, the receiver is synchronized by the $\overline{\text{DCD}}$ (Data Carrier Detect) input (Figure 9) and transfers successive bytes of data to the input of the Receiver FIFO. The single-sync-character mode requires that a match occur between the Sync Code Register and one incoming character before data transfer to the FIFO begins. The two-sync-character mode requires that two sync codes be received in sequence to establish synchronization. Subsequent to synchronization in any mode, data is accumulated in the shift register, and parity is optionally checked. An indication of parity error is carried through the Receiver FIFO with each character to the last empty location. Availability of a word at the FIFO output is indicated by the RDA status bit in the Status Register, as is a parity error (PE).

The SSDA and its internal registers are selected by RS, $\overline{\text{CS}}$, Read/Write (R/$\overline{\text{W}}$) and Enable control lines. To configure the SSDA, Control Registers are selected and the appropriate bits set. The Status Register is addressable for reading status.

Other I/O lines, in addition to Clear-to-Send ($\overline{\text{CTS}}$) and Data Carrier Detect ($\overline{\text{DCD}}$), include SM/DTR (Sync Match/Data Terminal Ready) and Transmitter Underflow (TUF). The transmitter and receiver each have individual clock inputs allowing simultaneous operation under separate clock control. Signals to the microprocessor are the Data Bus and Interrupt Request ($\overline{\text{IRQ}}$).

INITIALIZATION

During a power-on sequence, the SSDA is reset via the $\overline{\text{RESET}}$ input and internally latched in a reset condition to prevent erroneous output transitions. The Receiver Shift Register is set to all "1's". The Sync Code Register, Control Register 2, and Control Register 3 should be programmed prior to the programmed release of the Transmitter and/or Receiver Reset bits; these bits in Control Register 1 should be cleared after the $\overline{\text{RESET}}$ line has gone high.

TRANSMITTER OPERATION

Data is transferred to the transmitter section in parallel form by means of the data bus and Transmit Data FIFO. The Transmit Data FIFO is a 3-byte register whose status is indicated by the Transmitter Data Register Available status bit (TDRA) and its associated interrupt enable bit. Data is transferred through the FIFO on negative edges of Enable (E) pulses. Two data transfer modes are provided in the SSDA. The 1-byte transfer mode provides for writing data to the transmitter section (and reading from the receiver section) one byte at a time. The 2-byte transfer mode provides for writing two data characters in succession.

Data will automatically transfer from the last register location in the Transmit Data FIFO (when it contains data) to the Transmitter Shift Register during the last half of the last bit of the previous character. A character is transferred into the Shift Register by the Transmitter Clock. Data is transmitted *LSB first*, and odd or even parity can be optionally appended. The unused bit positions in short word length characters, from the data bus, are "don't cares". (Note: The data bus inputs may be reversed for applications requiring the MSB to be transferred first, e.g., IBM format for floppy disks; however, care must be taken to properly program the control registers — Table 1 will have its bit positions reversed.)

When the Shift Register becomes empty, and data is *not* available for transfer from the Transmit Data FIFO, an "underflow" occurs, and a character is inserted into the transmitter data stream to maintain character synchronization. The character transmitted on underflow will be either a "Mark" (all "1's") or the contents of the Sync Code Register, depending upon the state of the Transmit Sync Code on Underflow control bit. The underflow condition is indicated by a pulse (=1 Tx CLK high period) on the Underflow output (when in Tx Sync on underflow mode). The Underflow output occurs coincident with the transfer of the last half of the last bit preceding the underflow character. The Underflow status bit is set until cleared by means of the Clear Underflow control bit. This output may be used in floppy disk systems to synchronize write operations and for appending CRCC.

Transmission is initiated by clearing the Transmitter Reset bit in Control Register 1. When the Transmitter Reset bit is cleared, the first *full* positive half-cycle of the Transmit Clock will initiate the transmit cycle, with the transmission of data or underflow characters beginning on the negative edge of the Transmit Clock pulse which started the cycle. If the Transmit Data FIFO was not loaded, an underflow character will be transmitted (see Figure 4).

The Clear-to-Send ($\overline{\text{CTS}}$) input provides for automatic control of the transmitter by means of external system hardware; e.g., the modem $\overline{\text{CTS}}$ output provides the control in a data communications system. The $\overline{\text{CTS}}$ input resets and inhibits the transmitter section when high, but does not reset the Transmit Data FIFO. The TDRA status bit is inhibited by $\overline{\text{CTS}}$ being high in either the one-sync character or two-sync character mode of operation. In the external sync mode, TDRA is unaffected by $\overline{\text{CTS}}$ in order to provide Transmit Data FIFO status for preloading and operating the transmitter under the control of the $\overline{\text{CTS}}$ input. When the Transmitter Reset bit (Tx Rs) is set, the Transmit Data FIFO is cleared and the TDRA status bit is cleared. After one E clock has occurred, the Transmit Data FIFO becomes available for new data with TDRA inhibited.

RECEIVER OPERATION

Data and a presynchronized clock are provided to the SSDA receiver section by means of the Receive Data (Rx Data) and Receive Clock (Rx CLK) inputs. The data is a continuous stream of binary data bits without means for identifying character boundaries within the stream. It is, therefore, necessary to achieve character synchronization for the data at the *beginning* of the data block. Once synchronization is achieved, it is assumed to be retained for all successive characters within the block.

Data communications systems utilize the detection of sync codes during the initial portion of the preamble to establish character synchronization. This requires the detection of a single code or two successive sync codes. Floppy disk and cartridge tape units require sixteen bits of defined preamble and cassettes require eight bits of preamble to establish the reference for the start of record. All three are functionally equivalent to the detection of sync codes. Systems which do not utilize code detection techniques require custom logic external to the SSDA for character synchronization and use of the parallel-to-serial (external sync) mode. (Note: The Receiver Shift Register is set to ones when reset.)

SYNCHRONIZATION

The SSDA provides three operating modes with respect to character synchronization: one-sync-character mode, two-sync-character mode, and external sync mode. The external sync mode requires synchronization and control of the receiving section through the Data Carrier Detect ($\overline{\text{DCD}}$) input (see Figure 7). This external synchronization could consist of direct line control from the transmitting end of the serial data link or from external logic designed to detect the start of the message block. The one-sync-character mode searches on a bit-by-bit basis until a match is achieved between the data in the Shift Register and the Sync Code Register. The match indicates character synchronization is complete and will be retained for the message block. In the two-sync-character mode, the receiver searches for the first sync code match on a bit-by-bit basis and then looks for a second *successive* sync code character prior to establishing character synchronization. If the second sync code character is not received, the bit-by-bit search for the first sync code is resumed.

Sync codes received prior to the completion of synchronization (one or two character) are not transferred to the Receive Data FIFO. Redundant sync codes during the preamble or sync codes which occur as "fill characters" can automatically be stripped from the data, when the Strip Sync control bit is set, to minimize system loading The character synchronization will be retained until cleared by means of the Clear Sync bit, which also inhibits synchronization search when set

RECEIVING DATA

Once synchronization has been achieved, subsequent characters are automatically transferred into the Receive Data FIFO and clocked through the FIFO to the last empty location by E pulses (MPU System $\phi 2$). The Receiver Data Available status bit (RDA) indicates when data is available to be read from the last FIFO location (#3) when in the 1-byte transfer mode. The 2-byte transfer mode causes the RDA status bit to indicate data is available when the last two FIFO register locations are full. Data being available in the Receive Data FIFO causes an interrupt request if the Receiver Interrupt Enable (RIE) bit is set. The MPU will then read the SSDA Status Register which will indicate that data is available for the MPU read from the Receive Data FIFO register. The IRQ and RDA status bits are reset by a read from the FIFO. If more than one character has been received and is resident in the Receive Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA and IRQ status bits will again be set The read data operation for the 2-byte transfer mode requires an intervening E clock between reads to allow the FIFO data to shift. Optional parity is automatically checked as data is received, and the parity status condition is maintained with each character until the data is read from the Receive Data FIFO. Parity errors will cause an interrupt request if the Error Interrupt Enable (EIE) has been set. The parity bit is not transferred to the data bus but must be checked in the Status Register. NOTE: In the 2-byte transfer mode, parity should be checked prior to reading the second byte, since a FIFO read clears the error bit.

Other status bits which pertain to the receiver section are Receiver Overrun and Data Carrier Detect ($\overline{\text{DCD}}$). The Overrun status bit is automatically set when a transfer of a character to the Receive Data FIFO occurs and the first register of the Receive Data FIFO is full. Overrun causes an interrupt if Error Interrupt Enable (EIE) has been set. The transfer of the overrunning character into the FIFO causes the previous character in the FIFO input register location to be lost. The Overrun status bit is cleared by reading the Status Register (when the overrun condition is present), followed by a Receive data FIFO Register read. Overrun cannot occur and be cleared without providing an opportunity to detect its occurrence via the Status Register.

A positive transition on the $\overline{\text{DCD}}$ input causes an interrupt if the EIE control bit has been set. The interrupt caused by $\overline{\text{DCD}}$ is cleared by reading the Status Register when the $\overline{\text{DCD}}$ status bit is high, followed by a Receive data FIFO read The $\overline{\text{DCD}}$ status bit will subsequently follow the state of the $\overline{\text{DCD}}$ input when it goes low.

INPUT/OUTPUT FUNCTIONS

SSDA INTERFACE SIGNALS FOR MPU

The SSDA interfaces to the MC6800 MPU with an 8-bit bi-directional data bus, a chip-select line, a register-select line, an interrupt-request line, read/write line, an enable line, and a reset line. These signals, in conjunction with the MC6800 VMA output, permit the MPU to have complete control over the SSDA.

SSDA Bi-Directional Data (D0-D7) — The bi-directional data lines (D0-D7) allow for data transfer between the SSDA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an SSDA read operation.

SSDA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers, clocks data to and from the SSDA, and moves data through the FIFO Registers

Read/Write (R/$\overline{\text{W}}$) – The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the SSDA's input/output data bus interface. When Read/Write is high (MPU read cycle), SSDA output drivers are turned on if the chip is selected and a selected register is read. When it is low, the SSDA output drivers are turned off and the MPU writes into a selected register. The Read/Write signal is also used to select read-only or write-only registers within the SSDA

Chip Select ($\overline{\text{CS}}$) – This high-impedance TTL-compatible input line is used to address the SSDA. The SSDA is selected when $\overline{\text{CS}}$ is low. VMA should be used in generating the $\overline{\text{CS}}$ input to insure that false selects will not occur Transfers of data to and from the SSDA are then performed under the control of the Enable signal, Read/Write, and Register Select

Register Select (RS) – The Register Select line is a high-impedance input that is TTL compatible A high level is used to select Control Registers C2 and C3, the Sync Code Register, and the Transmit/Receive Data Registers. A low level selects the Control 1 and Status Registers (see Table 1).

Interrupt Request ($\overline{\text{IRQ}}$) – Interrupt Request is a TTL compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The Interrupt Request remains low until cleared by the MPU.

$\overline{\text{RESET}}$ Input – The $\overline{\text{RESET}}$ input provides a means of resetting the SSDA from an external source In the low state, the $\overline{\text{RESET}}$ input causes the following.

1. Receiver Reset (Rx Rs) and Transmitter Reset (Tx Rs) bits are set causing both the receiver and transmitter sections to be held in a reset condition.
2. Peripheral Control bits PC1 and PC2 are reset to zero, causing the SM/$\overline{\text{DTR}}$ output to be high.
3. The Error Interrupt Enable (EIE) bit is reset.
4. An internal synchronization mode is selected.
5. The Transmitter Data Register Available (TDRA) status bit is cleared and inhibited.
6. The Receiver Shift Register is set to 1's.

When $\overline{\text{RESET}}$ returns high (the inactive state), the transmitter and receiver sections will remain in the reset state until the Receiver Reset and Transmitter Reset bits are cleared via the data bus under software control. The control Register bits affected by $\overline{\text{RESET}}$ (Rx Rs, Tx Rs, PC1, PC2, EIE, and E/I Sync) cannot be changed when $\overline{\text{RESET}}$ is low.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data.

Transmit Clock (Tx CLK) – The Transmit Clock input is used for the clocking of transmitted data. The transmitter shifts data on the negative transition of the clock.

Receive Clock (Rx CLK) – The Receive Clock input is used for clocking in received data. The clock and data must be synchronized externally. The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) – The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format.

Transmit Data (Tx Data) – The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The SSDA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Sync Match/Data Terminal Ready, Data Carrier Detect, and Transmitter Underflow.

Clear-to-Send ($\overline{\text{CTS}}$) – The $\overline{\text{CTS}}$ input provides a real-time inhibit to the transmitter section (the Tx Data FIFO is not disturbed) A positive $\overline{\text{CTS}}$ transition resets the Tx Shift Register and inhibits the TDRA status bit and its associated interrupt in both the one-sync-character and two-sync-character modes of operation. TDRA is not affected by the $\overline{\text{CTS}}$ input in the external sync mode.

The positive transition of $\overline{\text{CTS}}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{\text{CTS}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by writing a "1" in the Clear $\overline{\text{CTS}}$ bit in Control Register 3 or in the Transmitter Reset bit. The $\overline{\text{CTS}}$ status bit subsequently follows the $\overline{\text{CTS}}$ input when it goes low.

The $\overline{\text{CTS}}$ input provides character timing for transmitter data when in the external sync mode. Transmission is initiated on the negative transition of the first *full* positive clock pulse of the transmitter clock (Tx CLK) after the release of $\overline{\text{CTS}}$ (see Figure 6).

Data Carrier Detect ($\overline{\text{DCD}}$) — The $\overline{\text{DCD}}$ input provides a real-time inhibit to the receiver section (the Rx FIFO is not disturbed). A positive DCD transition resets and inhibits the receiver section except for the Receive FIFO and the RDRA status bit and its associated IRQ

The positive transition of $\overline{\text{DCD}}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{\text{DCD}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by reading the Status Register and then the Receiver FIFO, or by writing a "1" into the Receiver Reset bit. The $\overline{\text{DCD}}$ status bit subsequently follows the $\overline{\text{DCD}}$ input when it goes low. The $\overline{\text{DCD}}$ input provides character synchronization timing for the receiver during the external sync mode of operation. The receiver will be initialized and data will be sampled on the positive transition of the first *full* Receive Clock cycle after release of $\overline{\text{DCD}}$ (see Figure 7).

Sync Match/Data Terminal Ready (SM/$\overline{\text{DTR}}$) – The SM/$\overline{\text{DTR}}$ output provides four functions (see Table 1) depending on the state of the PC1 and PC2 control bits. When the Sync Match mode is selected (PC="1", PC2="0"), the output provides a one-bit-wide pulse when a sync code is detected. This pulse occurs for each sync code match even if the receiver has already attained synchronization. The SM output is inhibited when PC2="1". The $\overline{\text{DTR}}$ mode (PC1="0") provides an output level corresponding to the complement of PC2 ($\overline{\text{DTR}}$="0" when PC2="1"). (See Table 1.)

TABLE 1 — SSDA PROGRAMMING MODEL

Register	Control Inputs		Address Control		Register Content							
	RS	R/$\overline{\text{W}}$	AC2	AC1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Status (S)	0	1	X	X	Interrupt Request (IRQ)	Receiver Parity Error (PE)	Receiver Overrun (Rx Ovrn)	Transmitter Underflow (TUF)	Clear-to-Send ($\overline{\text{CTS}}$)	Data Carrier Detect ($\overline{\text{DCD}}$)	Transmitter Data Register Available (TDRA)	Receiver Data Available (RDA)
Control 1 (C1)	0	0	X	X	Address Control 2 (AC2)	Address Control 1 (AC1)	Receiver Interrupt Enable (RIE)	Transmitter Interrupt Enable (TIE)	Clear Sync	Strip Sync Characters (Strip Sync)	Transmitter Reset (Tx Rs)	Receiver Reset (Rx Rs)
Receive Data FIFO	1	1	X	X	D7	D6	D5	D4	D3	D2	D1	D0
Control 2 (C2)	1	0	0	0	Error Interrupt Enable (EIE)	Transmit Sync Code on Underflow (Tx Sync)	Word Length Select 3 (WS3)	Word Length Select 2 (WS2)	Word Length Select 1 (WS1)	1-Byte/2-Byte Transfer (1-Byte/2 Byte)	Peripheral Control 2 (PC2)	Peripheral Control 1 (PC1)
Control 3 (C3)	1	0	0	1	Not Used	Not Used	Not Used	Not Used	Clear Transmitter Underflow Status (CTUF)	Clear $\overline{\text{CTS}}$ Status (Clear $\overline{\text{CTS}}$)	One-Sync-Character/ Two-Sync Character Mode Control (1 Sync/ 2 Sync)	External/ Internal Sync Mode Control (E/I Sync)
Sync Code	1	0	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Transmit Data FIFO	1	0	1	1	D7	D6	D5	D4	D3	D2	D1	D0

X = Don't care

STATUS REGISTER

IRQ	Bit 7	The IRQ flag is cleared when the source of the IRQ is cleared. The source is determined by the enables in the Control Registers TIE, RIE, EIE.
	Bits 6–0	indicate the SSDA status at a point in time, and can be *reset* as follows
PE	Bit 6	Read Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0)
Rx Ovrn	Bit 5	Read Status and then Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0).
$\overline{\text{TUF}}$	Bit 4	A "1" into CTUF (C3 Bit 3) or into Tx Rs (C1 Bit 1)
$\overline{\text{CTS}}$	Bit 3	A "1" into Clear $\overline{\text{CTS}}$ (C3 Bit 2) or a "1" into Tx Rs (C1 Bit1)
$\overline{\text{DCD}}$	Bit 2	Read Status and then Rx Data FIFO or a "1" into Rx Rs (C1 Bit 0)
TDRA	Bit 1	Write into Tx Data FIFO.
RDA	Bit 0	Read Rx Data FIFO

CONTROL REGISTER 1

AC2, AC1	Bits 7, 6	Used to access other registers, as shown above
RIE	Bit 5	When "1", enables interrupt on RDA (S Bit 0)
TIE	Bit 4	When "1", enables interrupt on TDRA (S Bit 1)
Clear Sync	Bit 3	When "1", clears receiver character synchronization
Strip Sync	Bit 2	When "1", strips all sync codes from the received data stream.
Tx Rs	Bit 1	When "1", resets and inhibits the transmitter section
Rx Rs	Bit 0	When "1", resets and inhibits the receiver section

CONTROL REGISTER 3

CTUF	Bit 3	When "1", clears $\overline{\text{TUF}}$ (S Bit 4), and IRQ if enabled
Clear $\overline{\text{CTS}}$	Bit 2	When "1", clears $\overline{\text{CTS}}$ (S Bit 3), and IRQ if enabled.
1 Sync/2 Sync	Bit 1	When "1", selects the one-sync-character mode, when "0", selects the two-sync-character mode.
E/I Sync	Bit 0	When "1", selects the external sync mode, when "0", selects the internal sync mode.

CONTROL REGISTER 2

EIE	Bit 7	When "1", enables the PE, Rx Ovrn, TUF, $\overline{\text{CTS}}$, and $\overline{\text{DCD}}$ interrupt flags (S Bits 6 through 2).
Tx Sync	Bit 6	When "1", allows sync code contents to be transferred on underflow, and enables the TUF Status bit and output. When "0", an all mark character is transmitted on underflow.
WS3, 2, 1	Bits 5–3	Word Length Select

Bit 5 WS3	Bit 4 WS2	Bit 3 WS1	Word Length
0	0	0	6 Bits + Even Parity
0	0	1	6 Bits + Odd Parity
0	1	0	7 Bits
0	1	1	8 Bits
1	0	0	7 Bits + Even Parity
1	0	1	7 Bits + Odd Parity
1	1	0	8 Bits + Even Parity
1	1	1	8 Bits + Odd Parity

1-Byte/2-Byte	Bit 2	When "1", enables the TDRA and RDA bits to indicate when a 1-byte transfer can occur, when "0", the TDRA and RDA bits indicate when a 2-byte transfer can occur
PC2, PC1	Bits 1-0	SM/$\overline{\text{DTR}}$ Output Control

Bit 1 PC2	Bit 0 PC1	SM/$\overline{\text{DTR}}$ Output at Pin 5
0	0	1
0	1	Pulse ⎍ 1-Bit Wide, on SM
1	0	0
1	1	SM Inhibited, 0

NOTE When the SSDA is used in applications requiring the MSB of data to be received and transmitted first, the data bus inputs to the SSDA may be reversed (D0 to D7, etc.). Caution must be used when this is done since the bit positions in this table will be reversed, and the parity should not be selected.

Transmitter Underflow (TUF) – The Underflow output indicates the occurrence of a transfer of a "fill character" to the Transmitter Shift Register when the last location (#3) in the Transmit Data FIFO is emtpy. The Underflow output pulse is approximately one Tx CLK high period wide and occurs during the last half of the last bit of the character preceding the "Underflow" (see Figure 4). The Underflow output pulse does not occur when the Tx Sync bit is in the reset state.

SSDA REGISTERS

Seven registers in the SSDA can be accessed by means of the data bus. The registers are defined as read-only or write-only according to the direction of information flow. The Register Select input (RS) selects two registers in each state, one being read-only and the other write-only. The Read/Write input (R/$\overline{W}$) defines which of the two selected registers will actually be accessed. Four registers (two read-only and two write-only) can be accessed via the bus at any particular time. These registers and the required addressing are defined in Table 1.

CONTROL REGISTER 1 (C1)

Control Register 1 is an 8-bit write-only register that can be directly addressed from the data bus. Control Register 1 is accessed when RS = "0" and R/$\overline{W}$ = "0"

Receiver Reset (Rx Rs), C1 Bit 0 – The Receiver Reset control bit provides both a reset and inhibit function to the receiver section. When Rx Rs is set, it clears the receiver control logic, sync logic, error logic, Rx Data FIFO Control, Parity Error status bit, and $\overline{DCD}$ interrupt. The Receiver Shift Register is set to ones. The Rx Rs bit must be cleared after the occurrence of a low level on $\overline{RESET}$ in order to enable the receiver section of the SSDA.

Transmitter Reset (Tx Rs), C1 Bit 1 – The Transmitter Reset control bit provides both reset and inhibit to the transmitter section. When Tx Rs is set, it clears the transmitter control section, Transmitter Shift Register, Tx Data FIFO Control (the Tx Data FIFO can be reloaded after one E clock pulse), the Transmitter Underflow status bit, and the $\overline{CTS}$ interrupt, and inhibits the TDRA status bit (in the one-sync-character and two-sync-character modes).The Tx Rs bit must be cleared after the occurrence of a low level on $\overline{RESET}$ in order to enable the transmitter section of the SSDA. If the Tx FIFO is not preloaded, it must be loaded immediately after the Tx Rs release to prevent a transmitter underflow condition.

Strip Synchronization Characters (Strip Sync), C1 Bit 2 – If the Strip Sync bit is set, the SSDA will automatically strip all received characters which match the contents of the Sync Code Register. The characters used for synchronization (one or two characters of sync) are always stripped from the received data stream

Clear Synchronization (Clear Sync), C1 Bit 3 – The Clear Sync control bit provides the capability of dropping receiver character synchronization and inhibiting resynchronization. The Clear Sync bit is set to clear and inhibit receiver synchronization in *all* modes and is reset to zero to enable resynchronization.

Transmitter Interrupt Enable (TIE), C1 Bit 4 – TIE enables both the Interrupt Request output ($\overline{IRQ}$) and Interrupt Request status bit to indicate a transmitter service request. When TIE is set and the TDRA status bit is high, the $\overline{IRQ}$ output will go low (the active state) and the IRQ status bit will go high.

Receiver Interrupt Enable (RIE), C1 Bit 5 – RIE enables both the Interrupt Request output ($\overline{IRQ}$) and the Interrupt Request status bit to indicate a receiver service request. When RIE is set and the RDA status bit is high, the $\overline{IRQ}$ output will go low (the active state) and the IRQ status bit will go high.

Address Control 1 (AC1) and Address Control 2 (AC2), C1 Bits 6 and 7 – AC1 and AC2 select one of the write-only registers – Control 2, Control 3, Sync Code, or Tx Data FIFO – as shown in Table 1, when RS = "1" and R/W = "0".

CONTROL REGISTER 2 (C2)

Control Register 2 is an 8-bit write-only register which can be programmed from the data bus when the Address Control bits in Control Register 1 (AC1 and AC2) are reset, RS = "1" and R/$\overline{W}$ = "0".

Peripheral Control (PC1) and Peripheral Control 2 (PC2), C2 Bits 0 and 1 – Two control bits, PC1 and PC2, determine the operating characteristics of the Sync Match/$\overline{DTR}$ output PC1, when high, selects the Sync Match mode. PC2 provides the inhibit/enable control for the SM/$\overline{DTR}$ output in the Sync Match mode. A one-bit-wide pulse is generated at the output when PC2 is "0", and a match occurs between the contents of the Sync Code Register and the incoming data even if sync is inhibited (Clear Sync bit = "1"). The Sync Match pulse is referenced to the negative edge of Rx - CLK pulse causing the match (see Figure 3).

The Data Terminal Ready ($\overline{DTR}$) mode is selected when PC1 is low. When PC2 = "1" the SM/$\overline{DTR}$ output = "0" and vice versa. The operation of PC2 and PC1 is summarized in Table 1.

1-Byte/2-Byte Transfer (1-Byte/2-Byte), C2, Bit 2 – When 1-Byte/2-Byte is set, the TDRA and RDA status bits will indicate the availabitliy of their respective data FIFO registers for a single-byte data transfer. Alternately, if 1-Byte/2-Byte is reset, the TDRA and RDA status bits indicate when two bytes of data can be moved without a second status read. An intervening Enable pulse must occur between data transfers.

Word Length Selects (WS1, WS2, WS3), C2 Bits 3, 4, 5 – Word Length Select bits WS1, WS2, and WS3 select word lengths of 7, 8, or 9 bits including parity as shown in Table 1.

Transmit Sync Code on Underflow (Tx Sync), C2 Bit 6 – When Tx Sync is set, the transmitter will automatically send a sync character when data is not available for transmission. If Tx Sync is reset, the transmitter will transmit a Mark character (including the parity bit position) on underflow. When the underflow is detected, a pulse approximately one Tx CLK high period wide will occur on the underflow output if the Tx Sync bit is set. Internal parity generation is inhibited during underflow except for sync code fill character transmission in 8-bit plus parity word lengths.

Error Interrupt Enable (EIE), C2 Bit 7 – When EIE is set, the $\overline{\text{IRQ}}$ status bit will go high and the $\overline{\text{IRQ}}$ output will go low if:

1. A receiver overrun occurs. The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
2. $\overline{\text{DCD}}$ input has gone to a "1". The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
3. A parity error exists for the character in the last location (#3) of the Rx Data FIFO. The interrupt is cleared by reading the Rx Data FIFO.
4. The $\overline{\text{CTS}}$ input has gone to a "1". The interrupt is cleared by writing a "1" in the Clear $\overline{\text{CTS}}$ bit, C3 bit 2, or by a Tx Reset.
5. The transmitter has underflowed (in the Tx Sync on Underflow mode). The interrupt is cleared by writing a "1" into the Clear Underflow, C3 bit 3, or Tx Reset.

When EIE is a "0", the IRQ status bit and the $\overline{\text{IRQ}}$ output are disabled for the above error conditions. A low level on the $\overline{\text{RESET}}$ input resets EIE to "0".

CONTROL REGISTER 3 (C3)

Control Register 3 is a 4-bit write-only register which can be programmed from the data bus whe RS = "1" and R/$\overline{\text{W}}$ = "0" ,and Address Control bit AC1 = "1" and AC2 = "0".

External/Internal Sync Mode Conrol (E/I Sync), C3, Bit 0 – When the E/I Sync Mode bit is high, the SSDA is in the external sync mode and the receiver synchronization logic is disabled. Synchronization can be achieved by means of the $\overline{\text{DCD}}$ input or by starting Rx CLK at the midpoint of data bit 0 of a cahracter with $\overline{\text{DCD}}$ low. Both the transmitter and receiver sections operate as parallel – serial converters in the External Sync mode. The Clear Sync bit in Control Register 1 acts as a receiver sync inhibit when high to provide a bus controllable inhibit. The Sync Code Register can serve as a transmitter fill character register and a receiver match register in this mode. A "low" on the $\overline{\text{RESET}}$ input resets the E/I Sync Mode bit placing the SSDA in the internal sync mode.

One-Sync-Character/Two-Sync-Character Mode Control (1-Sync/2-Sync), C3 Bit 1 – When the 1-Sync/2-Sync bit is set, the SSDA will synchronize on a single match between the received data and the contents of the Sync Code Register. When the 1-Sync/2-Sync bit is reset, two *successive* sync characters must be received prior to receiver synhnchronization. If the second sync character is not detected, the bit-by-bit search resumes from the first bit in the second character. See the description of the Sync Code Register for more details.

Clear $\overline{\text{CTS}}$ Status (Clear $\overline{\text{CTS}}$), C3 Bit 2 – When a "1" is written into the Clear $\overline{\text{CTS}}$ bit, the stored status and interrupt are cleared. Subsequently, the $\overline{\text{CTS}}$ status bit reflects the state of the $\overline{\text{CTS}}$ input. The Clear $\overline{\text{CTS}}$ control bit does not affect the $\overline{\text{CTS}}$ input nor its inhibit of the transmitter section. The Clear $\overline{\text{CTS}}$ command bit is self-clearing, and writing a "0" into this bit is a nonfunctional operation.

Clear Transmit Underflow Status (CTUF), C3 Bit 3 – When a "1" is written into the CTUF status bit, the CTUF bit and its associated interrupt are reset. The CTUF command bit is self-clearing and writing a "0" into this bit is a nonfunctional operation.

SYNC CODE REGISTER

The Sync Code Register is an 8-bit register for storing the programmable sync code required for received data character synchronization in the one-sync-character and two-sync-character modes. The Sync Code Register also provides for stripping the sync/fill characters from the received data (a programmable option) as well as automatic insertion of fill characters in the transmitted data stream. The Sync Code Register is not utilized for receiver character synchronization in the external sync mode; however, it provides storage of receiver match and transmit fill characters.

The Sync Code Register can be loaded when AC2 and AC1 are a "1" and "0", respectively, and R/$\overline{\text{W}}$ = "0" and RS = "1".

The Sync Code Register may be changed after the detection of a match with the received data (the first sync code having been detected) to synchronize with a double-word sync pattern. (This sync code change must occur prior to the completion of the second character.) The sync match (SM) output can be used to interrupt the MPU system to indicate that the first eight bits have matched. The service routine would then change the sync match register to the second half of the pattern. Alternately, the one-sync-character mode can be used for sync codes for 16 or more bits by using software to check the second and subsequent bytes after reading them from the FIFO.

The detection of the sync code can be programmed to appear on the Sync Match/$\overline{\text{DTR}}$ output by writing a "1" in PC1 (C2 bit 0) and a "0" in PC2 (C2 bit 1). The Sync Match output will go high for one bit time beginning at the character interface between the sync code and the next character (see Figure 3).

PARITY FOR SYNC CHARACTER

Transmitter

Transmitter does not generate parity for the sync character except 9-bit mode.

9-bit (8-bit + parity)...8-bit sync character + parity
8-bit (7-bit + parity)...8-bit sync character (no parity)
7-bit (6-bit + parity)...7-bit sync character (no parity)

Receiver

At Synchronization

Receiver automatically strips the sync character(s) (two sync characters if '2 sync' mode is selected) which is used to establish synchronization. Parity is not checked for these sync characters.

After Synchronization Is Established

When 'strip sync' bit is selected, the sync characters (fill characters) are stripped and parity is not checked for the stripped sync (fill) characters. When "strip sync" bit is not selected (low), the sync character is assumed to be normal data and it is transferred into FIFO after parity checking. (When non-parity format is selected, parity is not checked.)

Strip Sync (C1, Bit 2)	**WS0-WS2 (Data Format) (C2, Bits 3-5)**	
1	X	No transfer of sync code No parity Check of sync code
0	With Parity	*Transfer data and sync codes Parity check
0	Without Parity	*Transfer data and sync codes No parity check

*Subsequent to synchronization

It is necessary to consider parity in the selected sync character in the following cases. Data Format is (6 + parity), (7 + parity), strip sync is not selected (low), and when sync code is used as a fill character after synchronization.

The transmitter sends a sync character without parity, but the receiver checks the parity as if it is normal data. Therefore, the sync character should be chosen to match the parity check selected for the receiver in this special case. See the following section for unused bit assignment in short-word length.

RECEIVE DATA FIRST-IN FIRST-OUT REGISTER (Rx Data FIFO)

The Receive Data FIFO Register consists of three 8-bit registers which are used for buffer storage of received data. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register. The transfer from register to register occurs on E pulses. The RDA status bit will be high when data is available in the last location of the Rx Data FIFO.

In an Overrun condition, the overrunning character will be transferred into the full first stage of the FIFO register and will cause the loss of that data character. Successive overruns continue to overwrite the first register of the FIFO. This destruction of data is indicated by means of the Overrun status bit The Overrun bit will be set when the overrun occurs and remains set until the Status Register is read, followed by a read of the Rx Data FIFO.

Unused data bits for short word lengths (including the parity bit) will appear as "0's" on the data bus when the Rx Data FIFO is read

TRANSMIT DATA FIRST-IN FIRST-OUT REGISTER (Tx Data FIFO)

The Transmit Data FIFO Register consists of thee 8-bit registers which are used for buffer storage of data to be transmitted. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register The transfer is clocked by E pulses

The TDRA status bit will be high if the Tx Data FIFO is available for data.

Unused data bits for short word lengths will be handled as "don't cares." The parity bit is not transferred over the data bus since the SSDA generates parity at transmission.

When an Underflow occurs, the Underflow character will be either the contents of the Sync Code Register or an all "1's" character. The underflow will be stored in the Status Register until cleared and will appear on the Underflow output as a pulse approximatley one Tx CLK high period wide.

STATUS REGISTER (S)

The Status Register is an 8-bit read-only register which provides the real-time status of the SSDA and the associated serial data channel. Reading the Status Register is a non-destructive process. The method of clearing status bits depends upon the function each bit represents and is discussed for each bit in the register.

Receiver Data Available (RDA), S Bit 0 – The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. The receiver data being present in the last register (#3) of the FIFO causes RDA to be high for the 1-byte transfer mode. The RDA bit being high indicates that the last two registers (#2 and #3) are full when in the 2-byte transfer mode. The second character can be read without a second status read (to determine that the character is available). An E pulse must occur between reads of the Rx Data FIFO to allow the FIFO to shift. Status must be read on a word-by-word basis if receiver data error checking is important The RDA status bit is reset automatically when data is not available.

Transmitter Data Register Available (TDRA), S Bit 1 – The TDRA status bit indicates that data can be loaded into the Tx Data FIFO Register. The first register (#1) of the Tx Data IFFO being empty will be indicated by a high level in the TDRA status bit in the 1-byte transfer mode. The first two registers (#1 and #2) must be empty for TDRA to be high when in the 2-byte transfer mode. The Tx Data FIFO can be loaded with two bytes without an intervening status read; however, one E pulse must occur between loads. TDRA is inhibited by the Tx Reset or $\overline{\text{RESET}}$. When Tx Reset is set, the Tx Data FIFO is cleared and then released on the next E clock pulse. The Tx Data FIFO can then be loaded with up to three characters of data, even though TDRA is inhibited. This feature allows preloading data prior to the release of Tx Reset. A high level on the $\overline{\text{CTS}}$ input inhibits the TDRA status bit in either sync mode of operation (one-sync-character or two-sync-character). $\overline{\text{CTS}}$ does not affect TDRA in the external sync mode. This enables the SSDA to operate under the control of the $\overline{\text{CTS}}$ input with TDRA indicating the status of the Tx Data FIFO. The $\overline{\text{CTS}}$ input does not clear the Tx Data FIFO in any operating mode.

Data Carrier Detect ($\overline{\text{DCD}}$), S Bit 2 – A positive transition on the $\overline{\text{DCD}}$ input is stored in the SSDA until cleared by reading both Status and Rx Data FIFO. A "1" written into Rx Rs also clears the stored $\overline{\text{DCD}}$ status. The $\overline{\text{DCD}}$ status bit, when set, indicates that the $\overline{\text{DCD}}$ input has gone high. The reading of Status followed by reading of the Receive Data FIFO allows Bit 2 of subsequent Status reads to indicate the state of the $\overline{\text{DCD}}$ input until the next positive transition.

Clear-to-Send ($\overline{CTS}$), S Bit 3 – A positive transition on the $\overline{CTS}$ input is stored in the SSDA until cleared by writing a "1" into the Clear $\overline{CTS}$ control bit or the Tx Rs bit The $\overline{CTS}$ status bit, when set, indicates that the $\overline{CTS}$ input has gone high The Clear $\overline{CTS}$ command (a "1" into C3 Bit 2) allows Bit 3 of subsequent Status reads to indicate the state of the $\overline{CTS}$ input until the next positive transition

Transmitter Underflow (TUF), S Bit 4 – When data is not available for the transmitter, an underflow occurs and is so indicated in the Status Register (in the Tx Sync on underflow mode). The underflow status bit is cleared by writing a "1" into the Clear Underflow (CTUF) control bit or the Tx Rs bit TUF indicates that a sync character will be transmitted as the next character A TUF is indicated on the output *only* when the contents of the Sync Code Register is to be transferred (transmit sync code on underflow = "1")

Receiver Overrun (Rx Ovrn), S Bit 5 – Overrun indicates data has been received when the Rx Data FIFO is full, resulting in data loss. The Rx Ovrn status bit is set when overrun occurs The Rx Ovrn status bit is cleared by reading Status followed by reading the Rx Data FIFO or by setting the Rx Rs control bit.

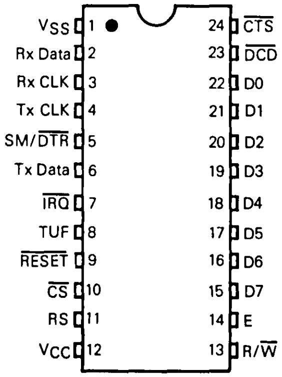
Receiver Parity Error (PE), S Bit 6 – The parity error status bit indicates that parity for the character in the last register of the Rx Data FIFO did not agree with selected parity The parity error is cleared when the character to which it pertains is read from the Rx Data FIFO or when Rx Rs occurs The $\overline{DCD}$ input does not clear the Parity Error or Rx Data FIFO status bits.

Interrupt Request (IRQ), S Bit 7 – The Interrupt Request status bit indicates when the $\overline{IRQ}$ output is in the active state ($\overline{IRQ}$ output = "0") The IRQ status bit is subject to the same interrupt enables (RIE, TIE, and EIE) as the $\overline{IRQ}$ output. The IRQ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip S Suffix	1.0	0°C to 70°C	MC6852S
	1.0	−40°C to 85°C	MC6852CS
	1.5	0°C to 70°C	MC68A52S
	1.5	−40°C to 85°C	MC68A520CS
	2.0	0°C to 70°C	MC68B52S
Plastic P Suffix	1.0	0°C to 70°C	MC6852P
	1.0	−40°C to 85°C	MC6852CP
	1.5	0°C to 70°C	MC68A52P
	1.5	−40°C to 85°C	MC68A52CP
	2.0	0°C to 70°C	MC68B52P

PIN ASSIGNMENTS



MC6854

Advanced Data-Link Controller (ADLC)

The MC6854 ADLC performs the complex MPU/data communication link function for the "Advanced Data Communication Control Procedure" (ADCCP), High-Level Data-Link Control (HDLC) and Synchronous Data-Link Control (SDLC) standards. The ADLC provides key interface requirements with improved software efficiency. The ADLC is designed to provide the data communications interface for both primary and secondary stations in stand-alone, polling, and loop configurations.

- M6800 Compatible
- Protocol Features
  - Automatic Flag Detection and Synchronization
  - Zero Insertion and Deletion
  - Extendable Address, Control and Logical Control Fields (Optional)
  - Variable Word Length Information Field — 5-, 6-, 7-, or 8-Bits
  - Automatic Frame Check Sequence Generation and Check
  - Abort Detection and Transmission
  - Idle Detection and Transmission
- Loop Mode Operation
- Loop Back Self-Test Mode
- NRZ/NRZI Modes
- Quad Data Buffers for Each Rx and Tx
- Prioritized Status Register (Optional)
- MODEM/DMA Loop Interface

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS

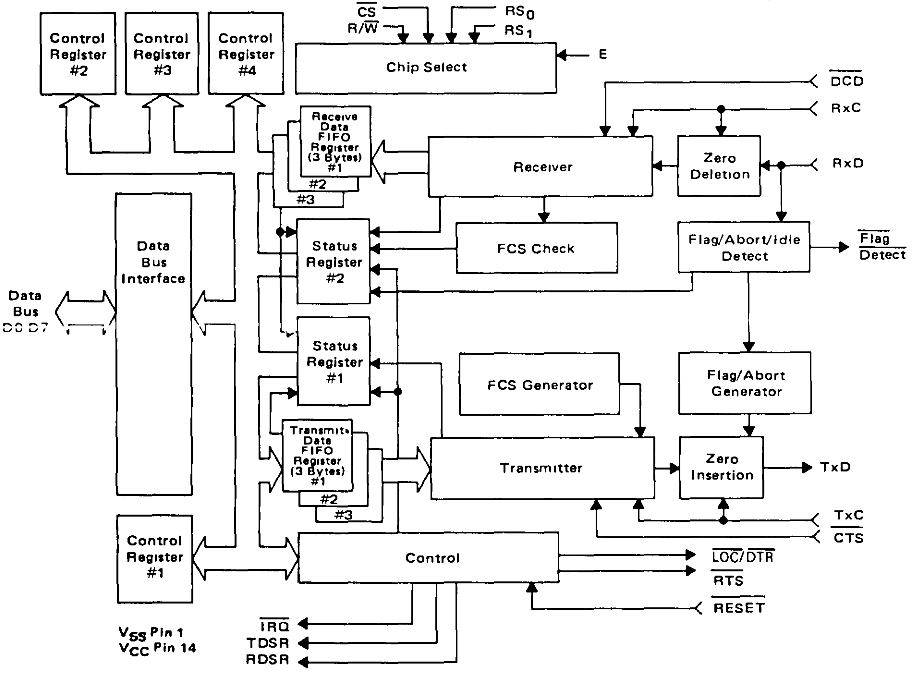
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6854, MC68A54, MC68B54 MC6854C, MC68A54C	T_A	(T_L to T_H) 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 115 65	°C/W

FIGURE 1 — ADLC GENERAL BLOCK DIAGRAM



CS
R/W
RS0
RS1
Chip Select
E
Control Register #2
Control Register #3
Control Register #4
DCD
RxC
Receive Data FIFO Register (3 Bytes) #1
#2
#3
Receiver
Zero Deletion
RxD
Data Bus Interface
Status Register #2
FCS Check
Flag/Abort/Idle Detect
Flag Detect
Data Bus D0-D7
Status Register #1
FCS Generator
Flag/Abort Generator
Transmit Data FIFO Register (3 Bytes) #1
#2
#3
Transmitter
Zero Insertion
TxD
TxC
CTS
Control Register #1
Control
LOC/DTR
RTS
RESET
IRQ
TDSR
RDSR
VSS Pin 1
VCC Pin 14

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K - (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5 0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2 0	–	–	V
Input Low Voltage		V_{IL}	–	–	V_{SS} + 0 8	V
Input Leakage Current (V_{in} = 0 to 5 25 V)	All Inputs Except D0-D7	I_{in}	–	1.0	2 5	µA
Hi-Z (Off-State) Input Current (V_{in} = 0 4 to 2 4 V, V_{CC} = 5 25 V)	D0-D7	I_{IZ}	–	2 0	10	µA
dc Output High Voltage (I_{Load} = – 205 µA) (V_{Load} = – 100 µA)	 D0-D7 All Others	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4	 – –	 – –	V
dc Output Low Voltage (I_{Load} = 1 6 mA))		V_{OL}	–	–	V_{SS} + 0 4	V
Output Leakage Current (Off State) (V_{OH} = 2 4 V)	$\overline{IRQ}$	I_{OZ}	–	1 0	10	µA
Internal Power Dissipation (measured at T_A = 0°C)		P_{INT}	–	–	850*	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)	D0-D7 All Other Inputs	C_{in}	– –	– –	12.5 7 5	pF
	$\overline{IRQ}$ All Others	C_{out}	– –	– –	5 0 10	pF

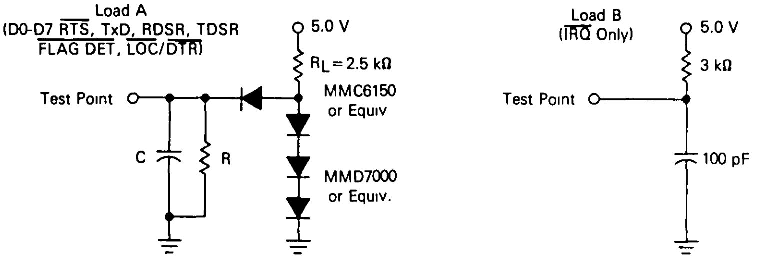
*For temperatures below 0°C, P_{INT} will increase

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	MC6854		MC68A54		MC68B54		Unit
		Min	Max	Min	Max	Min	Max	
Clock Pulse Width, Low (RxC, TxC)	PW_{CL}	700	–	450	–	280	–	ns
Clock Pulse Width, High (RxC, TxC)	PW_{CH}	700	–	450	–	280	–	ns
Serial Clock Frequency (RxC, TxC)	f_{SC}	–	0.66	–	1 0	–	1.5	MHz
Receive Data Setup Time	t_{RDSU}	150	–	100	–	50	–	ns
Receive Data Hold Time	t_{RDH}	60	–	60	–	60	–	ns
Request-to-Send Delay Time	t_{RTS}	–	680	–	460	–	340	ns
Clock-to-Data Delay for Transmitter	t_{TDD}	–	300	–	250	–	200	ns
Flag Detect Delay Time	t_{FD}	–	680	–	460	–	340	ns
$\overline{DTR}$ Delay Time	t_{DTR}	–	680	–	460	–	340	ns
Loop On-Line Control Delay Time	t_{LOC}	–	680	–	460	–	340	ns
RDSR Delay Time	t_{RDSR}	–	540	–	400	–	340	ns
TDSR Delay Time	t_{TDSR}	–	540	–	400	–	340	ns
Interrupt Request Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	µs
$\overline{RESET}$ Pulse Width	t_{RESET}	1.0	–	0.65	–	0.40	–	µs
Input Rise and Fall Times (Except Enable) (0 8 V to 2.0 V)	t_r, t_f	–	1.0*	–	1.0*	–	1 0*	µs

*1 0 µs or 10% of the pulse width, whichever is smaller

FIGURE 2 — BUS TIMING TEST LOADS


Load A
(D0-D7 RTS, TxD, RDSR, TDSR
FLAG DET, LOC/DTR)
5.0 V
R_L = 2.5 kΩ
MMC6150
or Equiv
Test Point
C
R
MMD7000
or Equiv.
Load B
(IRQ Only)
5.0 V
3 kΩ
Test Point
100 pF

C = 130 pF for D0-D7
= 30 pF for others

R = 11.7 kΩ for D0-D7
= 24 kΩ for others

FIGURE 3 — RECEIVER DATA SETUP/HOLD, FLAG DETECT AND LOOP ON-LINE CONTROL DELAY TIMING

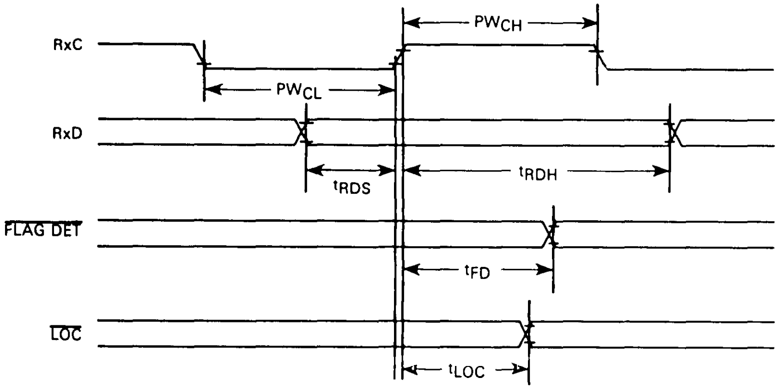
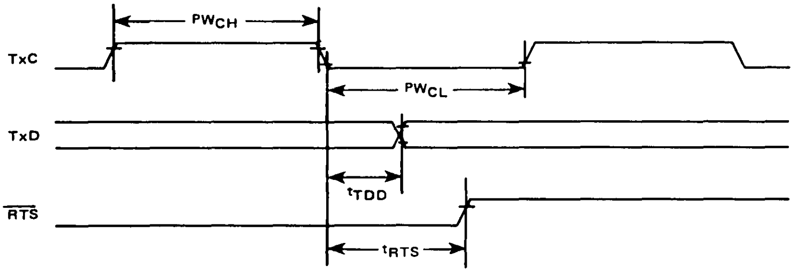
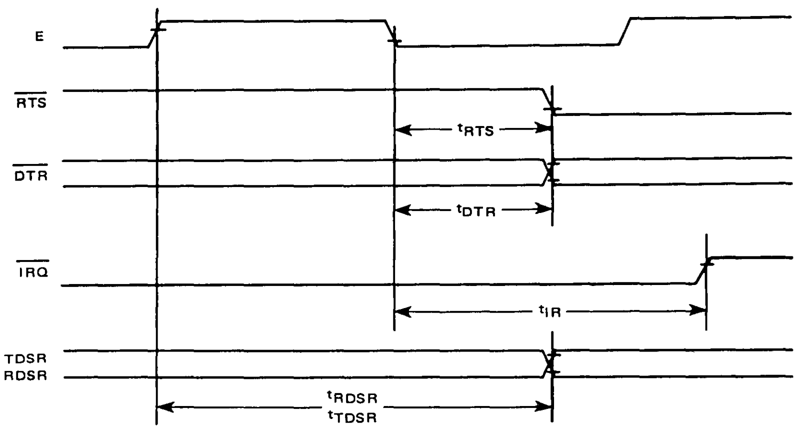

RxC
PW_{CH}
PW_{CL}
RxD
t_{RDS}
t_{RDH}
FLAG DET
t_{FD}
LOC
t_{LOC}

FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND REQUEST-TO-SEND DELAY TIMING


TxC
PW_{CH}
PW_{CL}
TxD
t_{TDD}
RTS
t_{RTS}

NOTE Timing measurements are referenced to and from a low voltage of 0 8 volts and a high voltage of 2 0 volts, unless otherwise noted

FIGURE 5 — TDSR/RDSR DELAYS, $\overline{IRQ}$ RELEASE DELAY, $\overline{RTS}$ AND $\overline{DTR}$ DELAY TIMING


E
$\overline{RTS}$
t_{RTS}
$\overline{DTR}$
t_{DTR}
$\overline{IRQ}$
t_{IR}
TDSR
RDSR
t_{RDSR}
t_{TDSR}

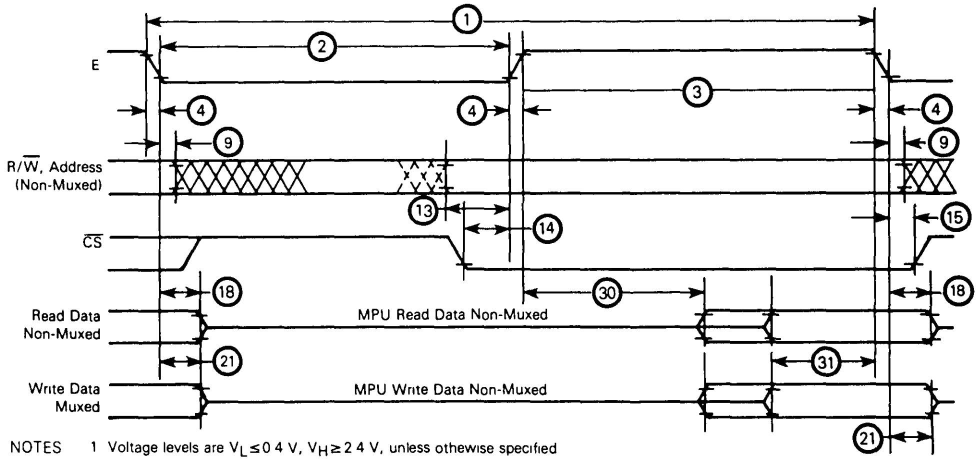
NOTE Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6854		MC68A54		MC68B54		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance)

FIGURE 6 – BUS TIMING

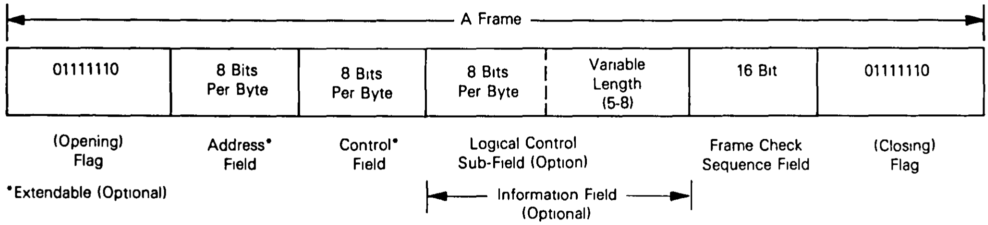

①
②
③
④
④
④
⑨
⑨
⑬
⑭
⑮
⑱
⑱
㉑
㉑
㉚
㉛
E
R/$\overline{W}$, Address (Non-Muxed)
$\overline{CS}$
Read Data Non-Muxed
MPU Read Data Non-Muxed
Write Data Muxed
MPU Write Data Non-Muxed

NOTES 1 Voltage levels are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless othewise specified
2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FRAME FORMAT

The ADLC transmits and receives data (information or control) in a format called a frame. All frames start with an opening flag (F) and end with a closing flag (F). Between the opening flag and closing flag, a frame contains an address field, control field, information field (optional) and frame check sequence field.

FIGURE 7 — DATA FORMAT OF A FRAME



Flag (F) — The flag is the unique binary pattern (01111110). It provides the frame boundary and a reference for the position of each field of the frame.

The ADLC transmitter generates a flag pattern internally and the opening flag and closing flags are appended to a frame automatically. Two successive frames can share one flag for a closing flag of the first frame and for the opening flag of the next frame, if the "FF"/"F" control bit in the control register is reset.

The receiver searches for a flag on a bit-by-bit basis and recognizes a flag at any time. The receiver establishes the frame synchronization with every flag. The flags mark the frame boundary and reference for each field but they are not transferred to the Rx FIFO. The detection of a flag is indicated by the Flag Detect output and by a status bit in the status register.

Order of Bit Transmission — Address, control and information field bytes are transferred between the MPU and the ADLC in parallel by means of the data bus. The bit on D0 (data bus bit 0, pin 22) is serially transmitted first, and the first serially received bit is transferred to the MPU on D0. The FCS field is transmitted and received MSB first.

Address (A) Field — The 8 bits following the opening flag are the address (A) field. The A-field can be extendable if the Auto-Address Extend Mode is selected in control register #3. In the Address Extend Mode, the first bit (bit 0) in every address octet becomes the extend control bit. When the bit is "0", the ADLC assumes another address octet will follow, and when the bit is "1", the address extension is terminated. A "null" address (all "0's") does not extend. In the receiver, the Address Present status bit distinguishes the address field from other fields. When an address byte is available to be read in the receive FIFO register, the Address Present status bit is set and causes an interrupt (if enabled). The Address Present bit is set for every address octet when the Address Extend Mode is used.

Control (C) Field — The 8 bits following the address field is the control (link control) field. When the Extended Control Field bit in control register #3 is selected, the C-field is extended to 16 bits

Information (I) Field — The I-field follows the C-field and precedes the FCS field. The I-field contains "data" to be transferred but is not always necessarily contained in every frame. The word length of the I-field can be selected from 5 to 8 bits per byte by control bits in control register #4. The I-field will continue until it is terminated by the FCS and closing flag. The receiver has the capability to handle a "partial" last byte. The last information byte can be any word length between 1 and 8 bits. If the last byte in the I-field is less than the selected word length, the receiver will right justify the received bits, fill the remaining bits of the receiver shift register with zeros, and transfer a full byte to the Rx FIFO. Regardless of selected byte length, the ADLC will transfer 8 bits of data to the data bus. Unused bits for word lengths of 5, 6, and 7 will be zeroed.

Logical Control (LC) Field — When the Logical Control Field Select bit, in control register #3, is selected the ADLC separates the I-field into two sub-fields. The first sub-field is the Logical Control field and the following sub-field is the "data" portion of the I-field. The logical control field is 8 bits and follows the C-field, which is extendable by octets, if it is selected. The last bit (bit 7) is the extend control bit, and if it is a "1", the LC-field is extended one octet.

NOTE

Hereafter the word "Information field" or "I-field" is used as the data portion of the information field, and excludes the logical control field. This is done in order to keep the consistency of the meaning of "Information field" as specified in SDLC, HDLC, and ADCCP standards.

Frame Check Sequence (FCS) Field – The 16 bits preceding the closing flag is the FCS field. The FCS is the "cyclic redundancy check character (CRCC)." The polynomial $x^{16}+x^{12}+x^{5}+1$ is used both for the transmitter and receiver. Both the transmitter and receiver polynomial registers are initialized to all "1's" prior to calculation of the FCS. The transmitter calculates the FCS on all bits of the address, control, logical control (if selected), and information fields, and transmits the complement of the resulting remainder as FCS. The receiver performs the similar computation on all bits of the address, control, logical control (if selected), information, and received FCS fields and compares the result to F0B8 (Hexadecimal). When the result matches F0B8, the Frame Valid status bit is set in the status register. If the result does not match, the Error status bit is set. The FCS generation, transmission, and checking are performed automatically by the ADLC transmitter and receiver. The FCS field is not transferred to the Rx FIFO.

Invalid Frame – Any valid frames should have at least the A-field, C-field, and FCS field between the opening flag and the closing flag. When invalid frames are received, the ADLC handles them as follows:

1. A short frame which has less than 25 bits between flags – the ADLC ignores the short frame and its reception is not reported to the MPU.
2. A frame less than 32 bits between the flags, or a frame 32 bits or more with an extended A-field or C-field that is not completed. – This frame is transferred into the Rx FIFO. The FCS/IF Error status bit indicates the reception of the invalid frame at the end of the frame.
3. Aborted Frame – The frame which is aborted by receiving an abort or DCD failure is also an invalid frame. Refer to "Abort" and "DCD status bit".

Zero Insertion and Zero Deletion – The Zero insertion and deletion, which allows the content of the frame to be transparent, are performed by the ADLC automatically. A binary 0 is inserted by the transmitter after any succession of five "1's" within a frame (A, C, LC, I, and FCS field). The receiver deletes a binary 0 that follows successive five continuous "1's" within a frame.

Abort – The function of prematurely terminating a data link is called "abort." The transmitter aborts a frame by sending at least eight consecutive "1's" immediately after the Tx Abort control bit in control register #4 is set to a "1". (Tx FIFO is also cleared by the Tx Abort control bit at the same time.) The abort can be extended up to (at least) 16 consecutive "1's", if the Abort Extend control bit in the control register #4 is set when an abort is sent. This feature is useful to force mark idle transmission. Reception of seven or more consecutive "1's" is interpreted as an abort by the receiver. The receiver responds to a received abort as follows.

1. An abort in an "out of frame" condition – an abort during the idle or time fill has no meaning. The abort reception is indicated in the status register as long as the abort condition continues; but neither an interrupt nor a stored condition occurs. The abort indication disappears after 15 or more consecutive "1's" are received (Received Idle status is set.)
2. An abort "in frame" after less than 26 bits are received after an opening flag – under this condition, any field of the aborted frame has not transferred to the MPU yet. The ADLC clears the aborted frame data in the FIFO and clears flag synchronization. Neither an interrupt nor a stored status occurs. The status indication is the same as (1) above.
3. An abort "in frame" after 26 bits or more are received after an opening flag – under this condition, some fields of the aborted frame might have been transferred onto the data bus. The abort status is stored in the receiver status register and the data of the aborted frame in the ADLC is cleared. The synchronization is also cleared.

Idle and Time Fill – When the transmitter is in an "out of frame" condition (the transmitter is not transmitting a frame), it is in an idle state. Either a series of contiguous flags (time fill) or a mark idle (consecutive "1's" on a bit-by-bit basis) is selected for the transmission in an idle state by the Flag/Mark Idle control bit. When the receiver receives 15 or more consecutive "1's", the Receive Idle status bit is set and causes an interrupt. The flags and mark idle are not transferred to the Rx FIFO.

OPERATION

INITIALIZATION

During a power-on sequence, the ADLC is reset via the $\overline{\text{RESET}}$ input and internally latched in a reset condition to prevent erroneous output transitions. The four control registers must be programmed prior to the release of the reset condition. The release of the reset condition is performed via software by writing a "0" into the Rx RS control bit (receiver) and/or Tx RS control bit (transmitter). The release of the reset condition must be done after the $\overline{\text{RESET}}$ input has gone high.

At any time during operation, writing a "1" into the Rx RS control bit or Tx RS control bit causes the reset condition of the receiver or the transmitter.

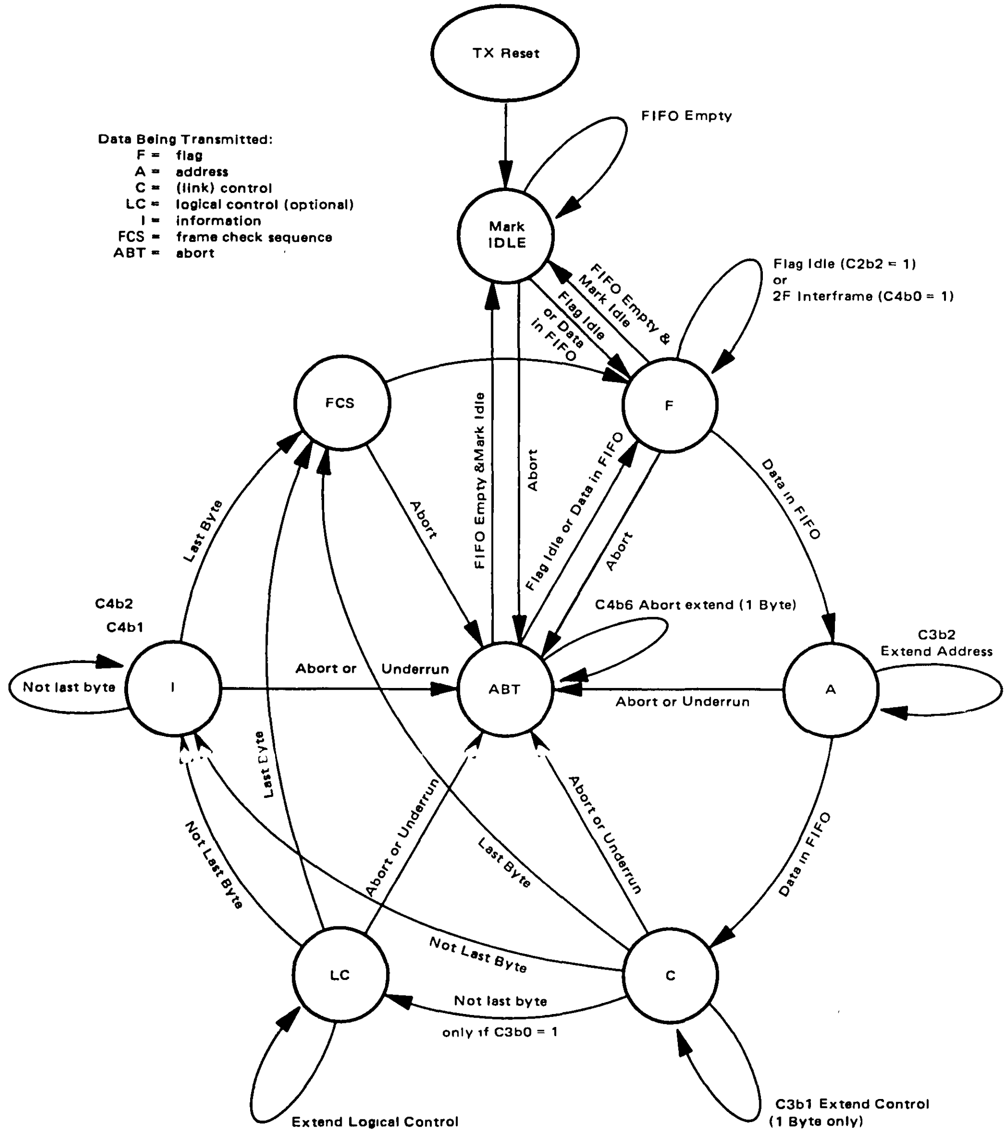
TRANSMITTER OPERATION

The Tx FIFO register cannot be pre-loaded when the transmitter is in a reset state. After the reset release, the Flag/Mark Idle control bit selects either the mark idle state (inactive idle) or the Flag "time fill" (active idle) state. This active or inactive mark idle state will continue until data is loaded into the Tx FIFO.

The availability of the Tx FIFO is indicated by the TDRA status bit under the control of the 2-Byte/1-Byte control bit. TDRA status is inhibited by the Tx RS bit or $\overline{\text{CTS}}$ input being high. When the 1-Byte mode is selected, one byte of the FIFO is available for data transfer when TDRA goes high. When the 2-Byte mode is selected, two successive bytes can be transferred when TDRA goes high.

The first byte (Address field) should be written into the Tx FIFO at the "Frame Continue" address. Then the transmission of a frame automatically starts. If the transmitter is in a mark idle state, the transfer of an address causes an opening flag within two or three transmitter clock cycles. If the transmitter has been in a time fill state, the current time fill flag being transmitted is assumed as an opening flag and the address field will follow it.

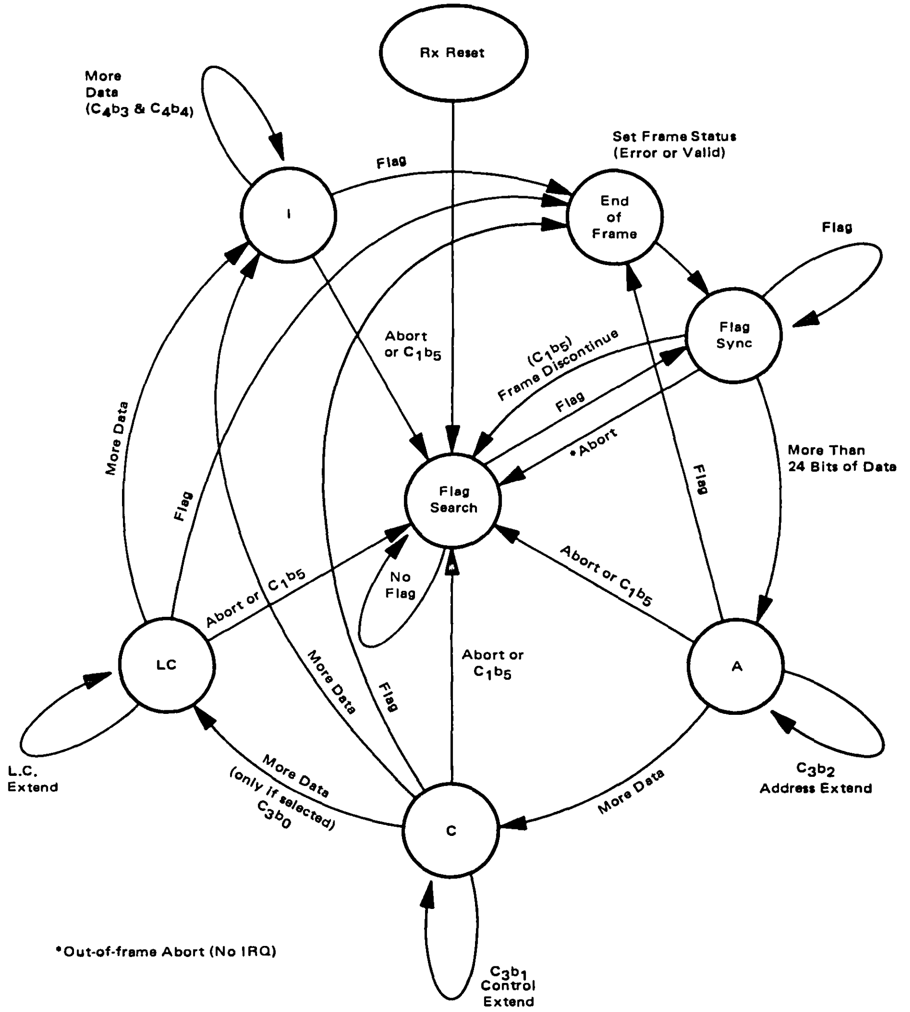
FIGURE 8a — ADLC TRANSMITTER STATE DIAGRAM
(Cibi refers to control register bit)



Data Being Transmitted:
F = flag
A = address
C = (link) control
LC = logical control (optional)
I = information
FCS = frame check sequence
ABT = abort

TX Reset
FIFO Empty
Mark IDLE
FIFO Empty & Mark Idle
Flag Idle or Data in FIFO
Flag Idle (C2b2 = 1) or 2F Interframe (C4b0 = 1)
FCS
F
FIFO Empty &Mark Idle
Abort
Flag Idle or Data in FIFO
Abort
Data in FIFO
Abort
Last Byte
C4b2
C4b1
C4b6 Abort extend (1 Byte)
C3b2 Extend Address
Not last byte
I
Abort or Underrun
ABT
Abort or Underrun
A
Last Byte
Abort or Underrun
Abort or Underrun
Data in FIFO
Not Last Byte
Last Byte
Not Last Byte
LC
Not last byte only if C3b0 = 1
C
Extend Logical Control
C3b1 Extend Control (1 Byte only)

FIGURE 8b — ADLC RECEIVER STATE DIAGRAM


Rx Reset
More Data (C_4b_3 & C_4b_4)
Flag
Set Frame Status (Error or Valid)
End of Frame
I
Flag
Flag Sync
Abort or C_1b_5
(C_1b_5) Frame Discontinue
Flag
•Abort
More Than 24 Bits of Data
More Data
Flag
Flag
Flag Search
Abort or C_1b_5
No Flag
Abort or C_1b_5
LC
A
Abort or C_1b_5
More Data
Flag
L.C. Extend
More Data (only if selected) C_3b_0
C
More Data
C_3b_2 Address Extend
C_3b_1 Control Extend

•Out-of-frame Abort (No IRQ)

A frame continues as long as data is written into the Tx FIFO at the "Frame Continue" address. The ADLC internally keeps track of the field sequence in the frame. The frame format is described in the "FRAME FORMAT" section.

The frame is terminated by one of two methods. The most efficient way to terminate the frames from a software standpoint is to write the last data character into the Transmit FIFO "Frame Terminate" address (RS1, RS0 = 11) rather than the Transmit FIFO "Frame Continue" address (RS1, RS0 = 10). An alternate method is to follow the last write of data in the Tx FIFO "Frame Continue" address with the setting of the Transmit Last Data control bit. Either method causes the last character to be transmitted and the FCS field to automatically be appended along with a closing flag. Data for a new frame can be loaded into the Tx FIFO immediately after the old frame data, if TDRA is high. The closing Flag can serve as the opening Flag of the next frame or separate opening and closing Flags may be transmitted. If a new frame is not ready to be transmitted, the ADLC will automatically transmit the Active (Flag) or Inactive (Mark) Idle condition.

If the Tx FIFO becomes empty at any time during frame transmission (the FIFO has no data to transfer into transmitter shift register during transmission of the last half of the next to last bit of a word), an underrun will occur and the transmitter automatically terminates the frame by transmitting an abort. The underrun state is indicated by the Tx Underrun status bit.

Any time the Tx ABORT Control bit is set, the transmitter immediately aborts the frame (transmits at least 8 consecutive "1's") and clears the Tx FIFO If the Abort Extend Control bit is set at the time, an idle (at least 16 consecutive "1's") is transmitted. An abort or idle in an "out of frame" condition can be useful to gain 8 or 16 bits of delay. (For an example, see "Programming Considerations.")

The $\overline{\text{CTS}}$ (Clear-to-Send) input and $\overline{\text{RTS}}$ (Request-to-Send) output are provided for a MODEM or other hardware interface.

The TDRA/FC status bit (when selected to be Frame Complete Status) can cause an interrupt upon frame completion (i.e., a flag or abort completion).

Details regarding the inputs and outputs, status bits, control bits, and FIFO operation are described in their respective sections.

RECEIVER OPERATION

Data and a pre-synchronized clock are provided to the ADLC receiver section by means of the Receive Data (RxD) and Receive Clock (RxC) inputs. The data is a continuous stream of binary bits with the characteristic that a maximum of five "1's" can occur in succession unless Abort, Flag, or Idling condition occurs. The receiver continuously (on a bit-by-bit basis) searches for Flags and Aborts.

When a flag is detected, the receiver establishes frame synchronization to the flag timing. If a series of flags is received, the receiver resynchronizes to each flag.

If the frame is terminated before the internal buffer time expires (the frame data is less than 25 bits after an opening flag), the frame is simply ignored. Noise on the data input (RxD) during time fill can cause this kind of invalid frame.

The received serial data enters a 32-bit shift register (clocked by RxC) before it is transferred into the Rx Data FIFO. Synchronization is established when a Flag is detected in the first eight locations of the shift register. Once synchronization has been achieved, data is clocked through to the last byte location of the shift register where it is transferred byte-per-byte into the Rx Data FIFO. The Rx Data FIFO is clocked by E to cause received data to move through the FIFO to the last empty register location. The Receiver Data Available status bit (RDA) indicates when data is present in the last register (Register #3) for the 1-Byte Transfer Mode. The 2-Byte Transfer Mode causes the RDA status bit to indicate data is available when the last two FIFO register locations (Registers #2 and #3) are full. If the data character present in the FIFO is an address octet, the status register will exhibit an Address Present status condition. Data being available in the Rx Data FIFO causes an interrupt to be initiated (assuming the receiver interrupt is enabled, RIE="1"). The MPU will read the ADLC Status Register as a result of the interrupt or in its turn in a polling sequence. RDA or Address Present will indicate that receiver data is available and the MPU should subsequently read the Rx Data FIFO register The interrupt and status bit will then be reset automatically. If more than one character had been received and was resident in the Rx Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA status bit and interrupt will again be SET. In the 2-Byte Transfer Mode both data bytes may be read on consecutive E cycles. Address Present provides for 1 byte transfers only.

The sequence of each field in the received frame is automatically handled by the ADLC. The frame format is described in the "FRAME FORMAT" section.

When a closing flag is received, the frame is terminated. The 16 bits preceding the closing flag are regarded as the FCS and are not transferred to the MPU. Whatever data is present in the most-significant byte portion of the receiver buffer register it is right justified and transferred to the Rx FIFO. The frame boundary pointer, which is explained in the "Rx FIFO REGISTER" section, is set simultaneously in the Rx FIFO. The frame boundary pointer sets the Frame Valid status bit (when the frame was completed with no error) or the FCS/IF Error Status bit (when the frame was completed with error) when the last byte of the frame appears at the last location of the Rx FIFO. As long as the Frame Valid or FCS/IF Error status bit is set, the data transfer from the second location of the Rx FIFO to the last location of the Rx FIFO is inhibited

Any time the Frame Discontinue control bit is set, the ADLC discards the current frame data in the ADLC without dropping flag synchronization. This feature can be used to ignore a frame which is addressed to another station.

The reception of an abort or idle is explained in the "FRAME FORMAT" section. The details regarding the inputs, outputs, status bits, control bits, and Rx FIFO operation are described in their respective sections.

LOOP MODE OPERATION

The ADLC in the loop mode, not only performs the transmission and receiving of data frames in the manner previously described, but also has additional features for gaining and relinquishing loop control In Figure 9a, a configuration is shown which depicts loop mode operation. The system configuration shows a primary station and several secondary stations. The loop is always under control of the primary station. When the primary wants to receive data, it transmits a Poll sequence and allows frame transmission to secondary stations on the loop Each secondary is in series and adds one bit of delay to the loop Secondary A in the figure receives data from the primary via its Rx Data Input, delays the data 1 bit, and transmits it to secondary B via its Tx Data Output. Secondaries B, C, and D operate in a similar manner. Therefore, data passes through each secondary and is received back by the primary controller.

Certain protocol rules must be followed in the manner by which the secondary station places itself on-loop (connects its transmitter output to the loop), goes active on the loop (starts transmitting its own station's data on the loop), and goes off the loop (disconnects its transmitter output). Otherwise loop data to other stations down loop would be interfered The data stream always flows the same way and the order in which secondary terminals are serviced is determined by the hardware configuration. The primary controller times the delay through the loop. Should it exceed n+1 bit times, where n is the number of secondary terminals on the loop, it will indicate a loop failure. Control is transferred to a secondary by transmitting a "Go Ahead" signal following the closing Flag of a polling frame (request for a response from the secondary) from the primary station. The "Go Ahead" from the primary is a "0" and seven "1's" followed by mark

FIGURE 9a — TYPICAL LOOP CONFIGURATION

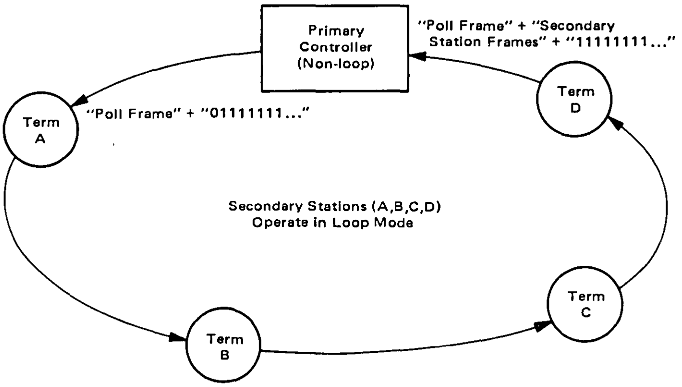
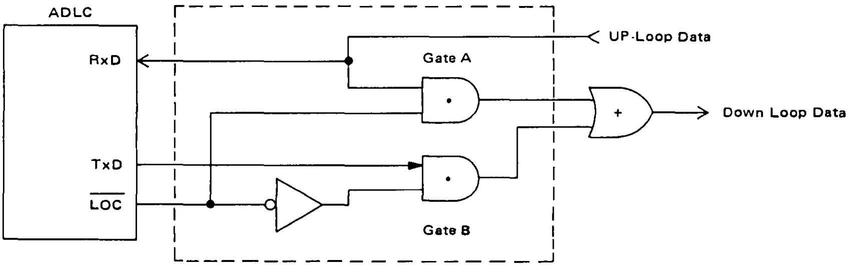

Primary Controller (Non-loop)
"Poll Frame" + "Secondary Station Frames" + "11111111..."
"Poll Frame" + "01111111..."
Term A
Term B
Term C
Term D
Secondary Stations (A,B,C,D) Operate in Loop Mode

FIGURE 9b — EXAMPLE OF EXTERNAL LOOP LOGIC



idling. The primary can abort its response request by interrupting its idle with flags. The secondary should immediately stop transmission and return control back to the primary. When the secondary completes its frame, a closing flag is transmitted followed by all "1's" The primary detects the final 01111111 . ("Go Ahead" to the primary) and control is given back to the primary. Note that, if a down-loop secondary (e.g., station D) needs to insert information following an up-loop station (e g., station A), the go ahead to station D is the last "0" of the closing flag from station A followed by "1's".

The ADLC in the primary station should operate in a non-loop full-duplex mode The ADLC in the secondaries should operate in a loop mode, monitoring up-loop data on its receiver data input. The ADLC can recognize the necessary sequences in the data stream to automatically go on/off the loop and to insert its own station data. The procedure is the following and is summarized in Table 1.

(1) Go On-Loop — When the ADLC powers up, the terminal station will be off line The first task is to become an active terminal on the loop. The ADLC must be connected to a Loop Link via an external switch as shown in Figure 9a. After a hardware reset, the ADLC $\overline{\text{LOC}}/\overline{\text{DTR}}$ Output will be in the high state and the up-loop receive data repeated through gate A to the down Loop stations. Any Up-Loop transmission will be received by the ADLC. The Loop Mode/Non-Loop Mode Control bit (bit 5 in Control Register 3) must be set to place the ADLC in the Loop Mode. The ADLC now monitors its Rx Data input for a string of seven consecutive "1's" which will allow a station to go on line. The Loop operation may be monitored by use of the Loop Status bit in Status Register 1. After power up and reset, this bit is a zero. When seven consecutive "1's" are received by the ADLC the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output will go to a low level, disabling gate A (refer to Figure 9b), enabling gate B and connecting the ADLC Tx Data output to the down Loop stations. The up Loop data is now repeated to the down Loop stations via the ADLC. A 1-bit delay is inserted in the data (in NRZI mode, there will be a 2-bit delay) as it circulates through the ADLC The ADLC is now on-line and the Loop Status bit in Status Register 1 will be at a one.

(2) Go Active after Poll — The receiver section will monitor the up-link data for a general or addressed poll command and the Tx FIFO should be loaded with data so that when the go ahead sequence of a zero followed by seven "1's" (01111111---) is detected, transmission can be initiated immediately. When the polling frame is detected, the Go-Active-On-Poll control bit must be set (bit 6 in Control

TABLE 1 — SUMMARY OF LOOP MODE OPERATION

STATE	RX SECTION	TX SECTION	LOOP STATUS BIT
OFF-LOOP	Rx section receives data from loop and searches for 7 "1's" (when On-Loop Control bit set) to go ON-LOOP.	Inactive 1) NRZ MODE. Tx data output is maintained "high" (mark). 2) NRZI MODE. Tx data output reflects the Rx data input state delayed by one bit time. (Not normally connected to loop.) The NRZI data is internally decoded to provide error-free transitions to On-Loop mode.	"0"
ON-LOOP	1) When Go-Active on poll bit is set, Rx section searches for 01111111 pattern (the EOP or 'Go Ahead') to become the active terminal on the loop. 2) When On-Loop control bit is reset, Rx section searches for 8 "1's" to go OFF-Loop.	Inactive 1) NRZ MODE. Tx data output reflects Rx data input state delayed one bit time. 2) NRZI MODE. Tx data output reflects Rx data input state delayed 2 bit times.	"1"
ACTIVE	Rx section searches for flag (an interrupt from the loop controller) at Rx data input. Received flag causes $\overline{\text{FD}}$ output to go low. IRQ is generated if $\overline{\text{RIE}}$ and FDSE control bits are set.	Tx data originates within ADLC until Go Active on Poll bit is reset and a flag or Abort is completed. Then returns to ON-Loop state.	"0"

Register 3). A maximum of seven bit times are available to set this control bit after the closing flag of the poll When the Go-Ahead is detected by the receiver, the ADLC will automatically change the seventh one to a zero so that the repeated sequence out gate B in Figure 9b is now an opening flag sequence (011111110). Transmission now continues from the Tx FIFO with data (address, control, etc.) as previously described. When the ADLC has gone active-on-poll, the Loop Status bit in Status Register 1 will go to a zero. The receiver searches for a flag, which indicates that the primary station is interrupting the current operation.

(3) Go Inactive when On-Loop — The Go-Active-On-Poll control bit may be $\overline{\text{RESET}}$ at any time during transmission. When the frame is complete (the closing Flag or abort is transmitted), the Loop is automatically released and the station reverts back to being just a 1-bit delay in the Loop, repeating up-link data. If the Go-Active-On-Poll control bit is not reset by software and the final frame is transmitted (Flag/Mark Idle bit = 0), then the transmitter will mark idle and will not release the loop to up-loop data. A Tx Abort command would have to be used in this case in order to go inactive when on the loop. Also, if the Tx FIFO was not preloaded with data (address, control, etc.) prior to changing the "Go Ahead Character" to a Flag, the ADLC will either transmit flags (active idle character) until data is loaded (when Flag/Mark Idle Control bit is high) or will go into an underrun condition and transmit an Abort (when Flag/Mark Idle control bit is low). When an abort is transmitted, the Go-Active-on-Poll control bit is reset automatically and the ADLC reverts to its repeating mode, (TxD = delayed RxD). When the ADLC transmitter lets go of the loop, the Loop Status bit will return to a "1", indicating normal on-loop retransmission of up-loop data

(4) Go Off-Loop — The ADLC can drop off the Loop (go off-line) similar to the way it went on-line When the Loop On-Line control bit is reset the ADLC receiver section looks for eight successive "1's" before allowing the $\overline{\text{LOC/DTR}}$ output to return high (the inactive state) Gate A in Figure 9b will be enabled and gate B disabled allowing the loop to maintain continuity without disturbance. The Loop Status bit will show an off-line condition (logical zero)

SIGNAL DESCRIPTIONS

All inputs of ADLC are high-impedance and TTL-compatible level inputs. All outputs of the ADLC are compatible with standard TTL. Interrupt Request ($\overline{\text{IRQ}}$), however, is an open-drain output (no internal pullup).

INTERFACE FOR MPU

Bidirectional Data Bus (D0-D7) — These data bus I/O ports allow the data transfer between ADLC and system bus. The data bus drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ADLC read operation.

Enable Clock (E) — E activates the address inputs ($\overline{\text{CS}}$, RS0, and RS1) and R/$\overline{\text{W}}$ input and enables the data transfer on the data bus. E also moves data through the Tx FIFO and Rx FIFO. E should be a free-running clock such as the MC6800 MPU system clock

Chip Select ($\overline{\text{CS}}$) — An ADLC read or write operation is enabled only when the $\overline{\text{CS}}$ input is low and the E clock input is high. (E•$\overline{\text{CS}}$)

Register Selects (RS0, RS1) – When the Register Select inputs are enabled by $(E \bullet \overline{CS})$, they select internal registers in conjunction with the Read/Write input and Address Control bit (control register 1, bit 0). Register addressing is defined in Table 2.

Read/Write Control Line ($R/\overline{W}$) – The $R/\overline{W}$ input controls the direction of data flow on the data bus when it is enabled by $(E \bullet \overline{CS})$. When $R/\overline{W}$ is high, the I/O Buffer acts as an output driver and as an input buffer when low. It also selects the Read Only and Write Only registers within the ADLC.

Reset Input ($\overline{RESET}$) – The $\overline{RESET}$ input provides a means of resetting the ADLC from a hardware source. In the "low state," the $\overline{RESET}$ input causes the following

- Rx Reset and Tx Reset are SET causing both the Receiver and Transmitter sections to be held in a reset condition
- Resets the following control bits: Transmit Abort, $\overline{RTS}$, Loop Mode, and Loop On-Line/$\overline{DTR}$.
- Clears all stored status condition of the status registers.
- Outputs. $\overline{RTS}$ and $\overline{LOC}/\overline{DTR}$ go high. TxD goes to the mark state ("1's" are transmitted).

When RESET returns "high" (the inactive state) the transmitter and receiver sections will remain in the reset state until Tx Reset and Rx Reset are cleared via the data bus under software control. The Control Register bits affected by $\overline{RESET}$ cannot be changed when $\overline{RESET}$ is "low."

Interrupt Request Output ($\overline{IRQ}$) – $\overline{IRQ}$ will be low if an interrupt situation exists and the appropriate interrupt enable has been set. The interrupt remains as long as the cause for the interrupt is present and the enable is set $\overline{IRQ}$ will be low as long as the $\overline{IRQ}$ status bit is set and is high if the $\overline{IRQ}$ status bit is not set.

CLOCK AND DATA OF TRANSMITTER AND RECEIVER

Transmitter Clock Input (TxC) – The transmitter shifts data on the negative transition of the TxC clock input When the Loop Mode or Test Mode is selected, TxC should be the same frequency and phase as the RxC clock. The data rate of the transmitter should not exceed the E frequency.

Receiver Clock Input (RxC) – The receiver samples the data on the positive transition of the RxC clock RxC should be synchronized with receive data externally

Transmit Data Output (TxD) – The serial data from the transmitter is coded in NRZ or NRZI (Zero Complement) data format.

Receiver Data Input (RxD) – The serial data to be received by the ADLC can be coded in NRZ or NRZI (Zero Complement) data format The data rate of the receiver should not exceed the E frequency If a partial byte reception is possible at the end of a frame, the maximum data rate of the receiver is indicated by the following relationship

$$f_{RxC} \leq \frac{1}{2t_E + 300 \text{ ns}}$$

where t_E is the period of E

PERIPHERAL/MODEM CONTROL

Request-to-Send Output ($\overline{RTS}$) – The Request-to-Send output is controlled by the Request-to-Send control bit in conjunction with the state of the transmitter section. When the $\overline{RTS}$ bit goes high, the $\overline{RTS}$ output is forced low. When the $\overline{RTS}$ bit returns low, the $\overline{RTS}$ output remains low until the end of the frame and there is no further data in the Tx FIFO for a new frame. The positive transition of $\overline{RTS}$ occurs after the completion of a Flag, an Abort, or when the RTS control bit is reset during a mark idling state When the $\overline{RESET}$ input is low, the $\overline{RTS}$ output goes high

Clear-to-Send Input ($\overline{CTS}$) – The $\overline{CTS}$ input provides a real-time inhibit to the TDRA status bit and its associated interrupt. The positive transition of $\overline{CTS}$ is stored within the ADLC to ensure its occurrence will be acknowledged by the system. The stored $\overline{CTS}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by writing a "1" in the Clear Tx Status bit or in the Transmitter Reset bit

Data-Carrier-Detect Input ($\overline{DCD}$)–The $\overline{DCD}$ input provides a real-time inhibit to the receiver section. A high level on the $\overline{DCD}$ input resets and inhibits the receiver register, but data in the Rx FIFO from a previous frame is not disturbed. The positive transition of $\overline{DCD}$ is stored within the ADLC to ensure that its occurrence will be acknowledged by the system. The stored $\overline{DCD}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by means of the Clear Rx Status Control bit or by the Rx Reset bit.

Loop On-Line Control/Data Terminal Ready Output ($\overline{LOC}/\overline{DTR}$) – The $\overline{LOC}/\overline{DTR}$ output serves as a $\overline{DTR}$ output in the non-loop mode or as a Loop Control output in the loop mode. When $\overline{LOC}/\overline{DTR}$ output performs the $\overline{DTR}$ function, it is turned on and off by means of the $\overline{LOC}/\overline{DTR}$ control bit When the $\overline{LOC}/\overline{DTR}$ control bit is high the $\overline{DTR}$ output will be low In the loop mode the $\overline{LOC}/\overline{DTR}$ output provides the means of controlling the external loop interface hardware to go On-line or Off-line. When the $\overline{LOC}/\overline{DTR}$ control bit is SET and the loop has "idled" for 7 bit times or more (RxD=01111111...), the $\overline{LOC}/\overline{DTR}$ output will go low (online) The $\overline{RESET}$ input being low will cause the $\overline{LOC}/\overline{DTR}$ output to be high.

Flag Detect Output ($\overline{FD}$) – An output to indicate the reception of a flag and initiate an external time-out counter for the loop mode operation The $\overline{FD}$ output goes low for 1 bit time beginning at the last bit of the flag character, as sampled by the receiver clock (RxC).

DMA INTERFACE

Receiver Data Service Request Output (RDSR) – The RDSR Output is provided primarily for use in DMA Mode operation and indicates (when high) that the Rx FIFO requests service (RSDR output reflects the RDA status bit regardless of the state of the RDSR mode control bit in CR1). If the prioritized Status Mode is selected, RDSR will be inhibited when any other receiver status conditions are present RDSR goes low when the Rx FIFO is read.

Transmitter Data Service Request Output (TDSR) — The TDSR Output is provided for DMA mode operation and indicates (when high) that the Tx FIFO request service regardless of the state of the TDSR Mode Control bit in CR1. TDSR goes low when the Tx FIFO is loaded. TDSR is inhibited by: the Tx RS control bit being SET, $\overline{\text{RESET}}$ being low, or CTS being high. If the prioritized status mode is used, Tx Underrun also inhibits TDSR. TDSR reflects the TDRA status bit except in the FC mode. In the FC mode the TDSR line is inhibited.

ADLC REGISTERS

Eight registers in the ADLC can be accessed by means of the MPU data and address buses. The registers are defined as read-only or write-only according to the direction of information flow. The addresses of these registers are defined in Table 2. The transitter FIFO register can be accessed by two different addresses, the "Frame Terminate" address and the "Frame Continue" address. (The function of these addresses are discussed in the FIFO section.)

TABLE 2 — REGISTER ADDRESSING

Register Selected	R/$\overline{\text{W}}$	RS1	RS0	Address Control Bit (C_1b_0)
Write Control Register #1	0	0	0	X
Write Control Register #2	0	0	1	0
Write Control Register #3	0	0	1	1
Write Transmit FIFO (Frame Continue)	0	1	0	X
Write Transmit FIFO (Frame Terminate)	0	1	1	0
Write Control Register #4	0	1	1	1
Read Status Register #1	1	0	0	X
Read Status Register #2	1	0	1	X
Read Receiver FIFO	1	1	X	X

RECEIVER DATA FIRST-IN FIRST-OUT REGISTER

Rx FIFO — The Rx FIFO consists of three 8-bit registers which are used for the buffer storage of received data Data bytes are always transferred from a full register to an adjacent empty register; and both phases of the E input clock are used for the data transfer. Each register has pointer bits which point the frame boundary. When these pointers appear at the last FIFO location, they update the Address Present, Frame Valid, or FCS/IF Error status bits.

The RDA status bit indicates the state of the Rx FIFO. When RDA status bit is "1", the Rx FIFO is ready to be read. The RDA status is controlled by the 2-Byte/1-Byte control bit When overrun occurs, the data in the first byte of the Rx FIFO are not longer valid

Both the Rx Reset bit and $\overline{\text{RESET}}$ input clear the Rx FIFO Abort ("in Frame") and a high level on the $\overline{\text{DCD}}$ input also clears the Rx FIFO, but the last bytes of the previous frame, which are separated by the frame boundary pointer, are not disturbed.

TRANSMITTER DATA FIRST-IN FIRST-OUT REGISTER

Tx FIFO — The Tx FIFO consists of three 8-bit registers which are used for buffer storage of data to be transmitted Data is always transferred from a full register to an empty adjacent register, the transfer occurs on both phases of the E input clock The Tx FIFO can be addressed by two different register addresses, the "Frame Continue" address and the "Frame Terminate" address Each register has pointer bits which point to the frame boundary. When a data byte is written at the "Frame Continue" address, the pointer of the first FIFO register is set When a data byte is written at the "Frame Terminate" address, the pointer of the first FIFO register is reset Rx RS control bit or Tx Abort control bit resets all pointers The pointer will shift through the FIFO When a positive transition is detected at the third location of FIFO, the transmitter initiates a frame with an open flag When the negative transition is detected at the third location of FIFO, the transmitter closes a frame, appending the FCS and closing Flag to the last byte

The Tx last control bit can be used instead of using the "Frame Terminate" address When the Tx last control bit is set with a "1", the logic searches the last byte location in the FIFO and resets the pointer in the FIFO register

The status of Tx FIFO is indicated by the TDRA status bit When TDRA is "1", the Tx FIFO is available for loading data The TDRA status is controlled by the 2-Byte/1-Byte control bit The Tx FIFO is reset by both Tx Reset and $\overline{\text{RESET}}$ input During this reset condition or when $\overline{\text{CTS}}$ input is high, the TDRA status bit is suppressed and data loading is inhibited

ADLC INTERNAL REGISTER STRUCTURE

		RS1 RS0 = 00	RS1 RS0 = 01	RS1 RS0 = 10	RS1 RS0 = 11
	Bit #	Status Register #1	Status Register #2	Receiver Data Register	
Read Only Registers	0	RDA	Address Present	Bit 0	Same as RS1, RS0 = 10
	1	Status #2 Read Request	Frame Valid	Bit 1	
	2	Loop	Inactive Idle Received	Bit 2	
	3	Flag Detected (When Enabled)	Abort Received	Bit 3	
	4	$\overline{\text{CTS}}$	FCS Error	Bit 4	
	5	Tx Underrun	$\overline{\text{DCD}}$	Bit 5	
	6	TDRA/Frame Complete	Rx Overrun	Bit 6	
	7	IRQ Present	RDA (Receiver Data Available)	Bit 7	

				Transmitter Data	Transmitter Data

CONTROL REGISTERS

CONTROL REGISTER 1 (CR1)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	TxRS	RxRS	Discontinue	TDSR Mode	RDSR Mode	TIE	RIE	AC
0	0	0	X								

b0 — Address Control (AC) — AC provides another RS (Register Select) signal internally. The AC bit is used in conjunction with RS0, RS1, and R/$\overline{W}$ inputs to select particular registers, as shown in Table 2.

b1 — Receiver Interrupt Enable (RIE) — RIE enables/disables the interrupt request caused by the receiver section. 1.. enable, 0...disable.

b2 — Transmitter Interrupt Eanble (TIE) — TIE enables/disables the interrupt request caused by the transmitter. 1...enable, 0...disable.

b3 — Receiver Data Service Request Mode (RDSR MODE) — The RDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When RDSR MODE is set, an interrupt request caused by RDA status is inhibited, and the ADLC does not request data transfer via the $\overline{IRQ}$ output

b4 — Transmitter Data Service Request Mode (TDSR MODE) — The TDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When TDSR MODE is set, an interrupt request caused by TDRA status is inhibited, and the ADLC does not request a data transfer via the $\overline{IRQ}$ output.

b5 — Rx Frame Discontinue (DISCONTINUE) — When the DISCONTINUE bit is set, the currently received frame is ignored and the ADLC discards the data of the current frame. The DISCONTINUE bit only discontinues the currently received frame and has no affect on subsequent frames, even if a following frame has entered the receiver section The DISCONTINUE bit is automatically reset when the last byte of the frame is discarded When the ignored frame is aborted by receiving an Abort or DCD failure, the DISCONTINUE bit is also reset

b6 — Receiver Reset (Rx RS) — When the Rx RS bit is "1", the receiver section stays in the reset condition. All reciever sections, including the Rx FIFO register and the receiver status bits in both status registers, are reset. (During reset, the stored DCD status is reset but the DCD status bit follows the $\overline{DCD}$ input.) Rx RS is set by forcing a low level on the $\overline{RESET}$ input or by writing a "1" into the bit from the data bus Rx RS must be reset by writing a "0" from the data bus after $\overline{RESET}$ has gone high

b7 — Transmitter Reset (Tx RS) — When the Tx RS bit is "1", the transmitter section stays in the reset condition and transmits marks ("1's"). All transmitter sections, including the Tx FIFO and the transmitter status bits, are reset (FIFO cannot be loaded) During reset, the stored CTS status is reset but the CTS status bit follows the $\overline{CTS}$ input. Tx RS is set by forcing a low level on the $\overline{RESET}$ input or by writing a "1" from the data bus. It must be reset by writing a "0" after $\overline{RESET}$ has gone high.

CONTROL REGISTER 2 (CR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				RTS	CLR TxST	CLR RxST	Tx Last	FC/TDRA Select	F/M Idle	2/1 Byte	PSE
0	1	0	0								

b0 — Prioritized Status Enable (PSE) — When the PSE bit is SET, the status bits in both status registers are prioritized as defined in the Status Register section When PSE is low, the status bits indicate current status without bit suppression by other status bits. The exception to this rule is the $\overline{\text{CTS}}$ status bit which always supresses the TDRA status.

b1 — 2-Byte/1-Byte Transfer (2/1 Byte) — When the 2/1 Byte bit is RESET the TDRA and RDA status bits then will indicate the availability of their respective data FIFO registers for a single-byte data transfer Similarly, if 2/1 Byte is set, the TDRA and RDA status bit indicate when two bytes of data can be moved without a second status read.

b2 — Flag/Mark Idle Select (F/M Idle) — The F/M Idle bit selects Flag characters or bit-by-bit Mark Idle for the time fill or the idle state of the transmitter. When Mark Idle is selected, Go-Ahead code can be generated for loop operation in conjunction with the 01/11 Idle control bit (C_3b_3) 1 .Flag time fill, 0...Mark Idle

b3 — Frame Complete/TDRA Select (FC/TDRA Select) — The FC/TDRA Select bit selects TDRA status or FC status for the TDRA/FC status bit indication. 1 .FC status, 0 TDRA status.

b4 — Transmit Last Data (Tx Last) — Tx Last bit provides another method to terminate a frame. This bit should be set after loading the last data byte and before the Tx FIFO empties. When the Tx Last bit is set, the ADLC assumes the byte is the last byte and terminates the frame by appending CRCC and a closing Flag This control bit is useful for DMA operation Tx Last bit automatically returns to the "0" state.

b5 — Clear Receiver Status (CLR Rx ST) — When a "1" is written into the CLR Rx ST bit, a reset signal is generated for the receiver status bits in status registers #1 and #2 (except AP and RDA bits). The reset signal is enabled only for the bits which have been present during the last "read status" operation. The CLR Rx ST bit automatically returns to the "0" state

b6 — Clear Transmitter Status (CLR Tx ST) — When a "1" is written into CLR Tx ST bit, a reset signal is generated for the transmitter status bits in status register #1 (except TDRA) The reset signal is enabled for the bits which have been present during the last "read status" operation. The CLR Tx ST bit automatically returns to the "0" state

b7 — Request-to-Send Control (RTS) — The RTS bit, when high, causes the $\overline{\text{RTS}}$ output to be low (the active state). When the RTS bit returns low and data is being transmitted, the $\overline{\text{RTS}}$ output remains low until the last character of the frame (the closing Flag or Abort) has been completed and the Tx FIFO is empty. If the transmitter is idling when the RTS bit returns low, the $\overline{\text{RTS}}$ output will go high (the inactive state) within two bit times.

CONTROL REGISTER 3 (CR3)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	LOC/	GAP/	Loop	FDSE	01/11	AEX	CEX	LCF
0	1	0	1	DTR	TST			Idle			

b0 — Logical Control Field Select (LCF) — The LCF select bit causes the first byte(s) of data belonging to the information field to remain 8-bit characters until the logical control field is complete. The logical control field (when selected) is an automatically extendable field which is extended when bit 7 of a logical control character is a "1." When the LCF Select bit is reset the ADLC assumes no logical control field is present for either the transmit or received data channels When the logical control field is terminated, the word length of the information data is then defined by WLS_1 and WLS_2.

b1 — Extended Control Field Select (C_{EX}) — When the C_{EX} bit is a "1", the control field is extended and asusmed to be 16 bits. When C_{EX} is "0", the control field is assumed to be 8 bits.

b2 — Auto/Address Extend Mode (A_{EX}) — The A_{EX} bit when "low" allows full 8 bits of the address octet to be utilized for addressing because address extension is inhibited. When the A_{EX} bit is "high," bit 0 of address octet equal to "0" causes the Address field to be extended by one octet. The exception to this automatic address field extension is when the first address octet is all "0's" (the Null Address).

b3 — 01/11 Idle (01/11 Idle) — The 01/11 Idle Control bit determines whether the inactive (Mark) idle condition begins with a "0" or not. If the 01/11 Idle Control is SET, the closing flag (or Abort) will be followed by a 011111...pattern. This is required of the controller for the "Go Ahead" character in the Loop Mode. When 01/11 is RESET, the idling condition will be all "1's".

b4 — Flag Detect Status Enable (FDSE) — The FDSE bit enables the FD status bit in Status Register #1 to indicate the occurrence of a received Flag character. The status indication will be accompanied by an interrupt if RIE is SET Flag detection will cause the Flag Detect output to go low for 1 bit time regardless of the state of FDSE

b5 — LOOP/NON-LOOP Mode (LOOP) — When the LOOP bit is set, loop mode operation is selected and the GAP/TST control bit, LOC/DTR control bit and LOC/DTR output are selected to perform the loop control functions When LOOP is reset, the ADLC operates in the point-to-point data communications mode

b6 — Go Active On Poll/Test (GAP/TST) — *In the Loop Mode* — The GAP/TST bit is used to respond to the poll sequence and to begin transmission When GAP/TST is set, the receiver searches for the "Go Ahead" (or End of Poll, EOP) The receiver "Go ahead" is converted to an opening Flag and the ADLC starts its own transmission. When GAP/TST is reset during the transmission, the end of the frame (the completion of Flag or Abort) causes the termination of the "go-active-on-poll" operation and the Rx Data to Tx Data link is re-established. The ADLC then returns to the "loop-on-line" state.

In the Non-Loop Mode — The GAP/TST bit is used for self-test purposes If GAP/TST bit is set, the TxD output is connected to the RxD input internally, and provides a "loop-back" feature. For normal operation, the GAP/TST bit should be reset.

b7 — Loop On-Line Control/DTR Control (LOC/DTR) — *In the Loop Mode* — The LOC/DTR bit is used to go on-line or to go off-line. When LOC/DTR is set, the ADLC goes to the on-line state after 7 consecutive "1's" occur at the RxD input. When LOC/DTR is reset, the ADLC goes to the "off-line" state after eight consecutive "1's" occur at the RxD input.

In the Non-Loop Mode — The LOC/DTR bit directly controls the Loop On-Line/$\overline{DTR}$ output state. 1...$\overline{DTR}$ output goes to low level, 0 $\overline{DTR}$ output goes to high level.

CONTROL REGISTER 4 (CR4)

				7	6	5	4	3	2	1	0
RS1	RS0	$R/\overline{W}$	AC	NRZI/NRZ	ABT_{EX}	ABT	Rx		Tx		"FF"/F
1	1	0	1				WLS_2	WLS_1	WLS_2	WLS_1	

b0 — Double Flag/Single Flag Interframe Control ("FF"/"F") — The "FF"/"F" Control bit determines whether the transmitter will transmit separate closing and opening Flags when frames are transmitted successively. When the "FF"/"F" control bit is low, the closing flag of the first frame will serve as the opening flag of the second frame. When the bit is high, independent opening and closing flags will be transmitted

b1, b2 — Transmitter Word Length Select (Tx WLS1 and WLS2) — Tx WLS1 and WLS2 are used to select the word length of the transmitter information field The encoding format is shown in Table 3.

b3, b4 — Receiver Word Length Select (Rx WLS1 and WLS2) — Rx WLS1 and WLS2 are used to select the word length of the receiver information field. The encoding format is shown in Table 3.

TABLE 3 — I-FIELD CHARACTER LENGTH SELECT

WLS_1	WLS_2	I-Field Character Length
0	0	5 bits
1	0	6 bits
0	1	7 bits
1	1	8 bits

b5 — Transmit Abort (ABT) — The ABT bit causes an Abort (at least 8 bits of "1" in succession) to be transmitted. The Abort is initiated and the Tx FIFO is cleared when the control bit goes high. Once Abort begins, the Tx Abort control bit assumes the low state.

b6 — Abort Extend (ABT_{EX}) — If ABT_{EX} is set, the abort code initiated by ABT is extended up to at least 16 bits of consecutive "1's", the mark Idle State.

b7 — NRZI (Zero Complement)/NRZ Select (NRZI/NRZ) — NRZI/NRZ bit selects the transmit/receive data format to be NRZI or NRZ in both Loop Mode or Non-Loop mode operation. When the NRZI Mode is selected, a 1-bit delay is added to the transmitted data (TxD) to allow for NRZI encoding 1. .NRZI, 0...NRZ.

NOTE

NRZI coding — The serial data remains in the same state to send a binary "1" and switches to the opposite state to send a binary "0"

STATUS REGISTER

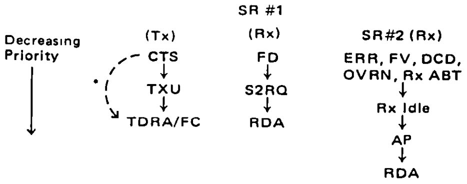
The Status Register #1 is the main status register. The IRQ bit indicates whether the ADLC requests service or not. The S2RQ bit indicates whether any bits in status register #2 request any service. TDRA and RDA, because they are most often used, are located in bit positions that are more convenient to test. RDA reflects the state of the RDA bit in status register #2.

The Status Register #2 provides the detailed status information contained in the S2RQ bit and these bits reflect receiver status. The FD bit is the only receiver status which is not indicated in status register #2.

The prioritized status mode provides maximum efficiency in searching the status bits and indicates only the most important action required to service the ADLC. The priority trees of both status registers are provided in Figure 10.

Reading the status register is a non-destructive process. The method of clearing status depends upon the bit's function and is discussed for each bit in the register.

FIGURE 10 — STATUS REGISTER PRIORITY TREE (PSE=1)



*Prioritized even when PSE = 0

NOTE. Status bit above will inhibit one below it.

STATUS REGISTER 1 (SR1)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{\text{W}}$	AC	IRQ	TDRA/FC	TXU	CTS	FD	LOOP	S2RQ	RDA
0	0	1	X								

b0 — Receiver Data Available (RDA) — The RDA status bit reflects the state of the RDA status bit in status register #2. It provides the means of achieving data transfers of received data in the full Duplex Mode without having to read both status registers.

b1 — Status Register #2 Read Request (S2RQ) — All the status bits (stored conditions) of status register #2 (except RDA bit) are logically ORed and indicated by the S2RQ status bit. Therefore, S2RQ indicates that status register #2 needs to be read. When S2RQ is "0", it is not necessary to read status register #2. The bit is cleared when the appropriate bits in status register #2 are cleared or when Rx Reset is used.

b2 — Loop Status (LOOP) — The LOOP status bit is used to monitor the loop operation of the ADLC This bit does not cause an IRQ. When Non-Loop Mode is selected, LOOP bit stays "0". When Loop Mode is selected, the LOOP status bit goes to "1" during "On-Loop" condition When ADLC is in an "Off-Loop" condition or "Go-Active-On-Poll" condition, the LOOP status bit is a "0"

b3 — Flag Detected (FD) — The FD Status bit indicates that a flag has been received if the Flag Detect Enable control bit has been set. The bit goes high at the last bit of the Flag Character received (when the Flag Detect Output goes low) and is stored until cleared by Clear Rx Status or Rx Reset

b4 — Clear-to-Send (CTS) — The $\overline{\text{CTS}}$ input positive transition is stored in the status register and causes an IRQ (if Enabled). The stored CTS condition and its IRQ are cleared by Clear Tx Status control bit or Tx Reset bit. After the stored status is reset, the CTS status bit reflects the state of the $\overline{\text{CTS}}$ input.

b5 — Transmitter Underrun (TxU) — When the transmitter runs out of data during a frame transmission, an underrun occurs and the frame is automatically terminated by transmitting an Abort. The underrun condition is indicated by the TxU status bit TxU can be cleared by means of the Clear Tx Status Control bit or by Tx Reset.

b6 — Transmitter Data Register Available/Frame Complete (TDRA/FC) — The TDRA Status bit serves two purposes depending upon the state of the Frame Complete/TDRA Select control bit. When this bit serves as a TDRA status bit, it indicates that data (to be transmitted) can be loaded into the Tx Data FIFO register The first register (Register #1) of the Tx Data FIFO being empty (TDRA = "1") will be indicated by the TDRA Status bit in the "1-Byte Transfer Mode." The first two registers (Registers #1 and #2) must be empty for TDRA to be high when in the "2-Byte Transfer Mode " TDRA is inhibited by Tx Reset, or $\overline{\text{CTS}}$ being high.

When the Frame Complete Mode of operation is selected, the TDRA/FC status bit goes high when an abort is transmitted or when a flag is transmitted with no data in the Tx FIFO. The bit remains high until cleared by resetting the TDRA/FC control bit or setting the Tx Reset bit.

b7 — Interrupt Request (IRQ) — The Interrupt Request status bit indicates when the $\overline{\text{IRQ}}$ output is in the active state ($\overline{\text{IRQ}}$ Output = "0"). The IRQ status bit is subject to the same interrupt enables (RIE, TIE) as the $\overline{\text{IRQ}}$ output, i e , with both transmitter and receiver interrupts enabled, the IRQ status bit is a logical ORed indication of Status Register 1 status bits. The IRQ bit only reflects the set status bits which have interrupts enabled. The IRQ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests.

STATUS REGISTER 2 (SR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				RDA	OVRN	DCD	ERR	Rx ABT	Rx Idle	FV	AP
0	1	1	X								

b0 — Address Present (AP) — The AP status bit provides the frame boundary and indicates an Address octet is available in the Rx Data FIFO register. In the Extended Addressing Mode, the AP bit continues to indicate addresses until the Address field is complete. The Address present status bit is cleared by reading data or by Rx Reset.

b1 — Frame Valid (FV) — The FV status bit provides the frame boundary indication to the MPU and also indicates that a frame is complete with no error. The FV status bit is set when the last data byte of a frame is transferred into the last location of the Rx FIFO (available to be read by MPU). Once FV status is set, the ADLC stops further data transfer into the last location of the Rx FIFO (in order to prevent the mixing of two frames) until the status bit is cleared by the Clear Rx Status bit or Rx Reset

b2 — Inactive Idle Received (Rx Idle) — The Rx Idle status bit indicates that a minimum of 15 consecutive "1's" have been received The event is stored within the status register and can cause an interrupt. The interrupt and stored condition are cleared by the Clear Rx Status Control bit The Status bit is the Logical OR of the receiver idling detector (which continues to reflect idling until a "0" is received) and the stored inactive idle condition.

b3 — Abort Received (RxABT) — The RxABT status bit indicates that seven or more consecutive "1's" have been received Abort has no meaning under out-of-frame conditions; therefore, no interrupt nor storing of the status will occur unless a Flag has been detected prior to the Abort. An Abort Received when "in frame" is stored in the status register and causes an IRQ The status bit is the logical OR of the stored conditions and the Rx Abort detect logic, which is cleared after 15 consecutive "1's" have occurred. The stored Abort condition is cleared by the Clear Rx Status Control bit or Rx Reset

b4 — Frame Check Sequence/Invalid Frame Error (ERR) — When a frame is complete with a cyclic redundancy check (CRC) error or a short frame error (the frame does not have complete Address and Control fields), the ERR status bit is set instead of the Frame Valid status bit. Other functions, frame boundry indication and control function, are exactly the same as for the Frame Valid status bit. Refer to the FV status bit.

b5 — Data Carrier Detect (DCD) — A positive transition on the $\overline{\text{DCD}}$ input is stored in the status register and causes an IRQ (if enabled) The stored DCD condition and its IRQ are cleared by the Clear Rx Status Control bit or RX Reset. After stored status is reset, the DCD status bit follows the state of the input Both the stored DCD condition and the $\overline{\text{DCD}}$ input cause the reset of the receiver section when they are high

b6 — Receiver Overrun (OVRN) — OVRN status indicates that receiver data has been transferred into the Rx FIFO when it is full, resulting in data loss The OVRN status is cleared by the Clear Rx Status bit or Rx Reset Continued overrunning only destroys data in the first FIFO Register.

b7 — Receiver Data Available (RDA) — The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. When the prioritized status mode is used, the RDA bit indicates that non-address and non-last data are available in the Rx FIFO. The receiver data being present in the last register of the FIFO causes RDA to be high for the "1-Byte Transfer Mode." The RDA bit being high indicates that the last two registers are full when in the "2-Byte Transfer Mode." The RDA status bit is reset automatically when data is not available.

PROGRAMMING CONSIDERATIONS

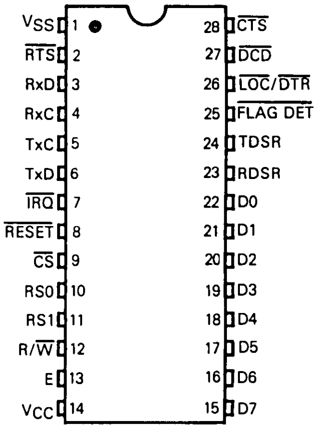
1. **Status Priority** — When the prioritized status mode is used, it is best to test for the lowest priority conditions first. The lowest priority conditions typically occur more frequently and are the most likely conditions to exist when the processor is interrupted.
2. **Stored vs Present Status** — Certain status bits (DCD, CTS, Rx Abort, and Rx Idle) indicate a status which is the logical OR of a stored and a present condition. It is the stored status that causes an interrupt and which is cleared by a Status Clear control bit. After being cleared, the status register will reflect the present condition of an input or a receiver input sequence.
3. **Clearing Status Registers** — In order to clear an interrupt with the two Status Clear control bits, a particular status condition must be read before it can be cleared. In the prioritized mode, clearing a higher priority condition might result in another $\overline{IRQ}$ caused by a lower priority condition whose status was suppressed when a status register was first read. This guarantees that a status condition is never inadvertently cleared.
4. **Clearing the Rx FIFO** — An Rx Reset will effectively clear the contents of all three Rx FIFO bytes However, the FIFO may contain data from two different frames when abort or DCD failure occurs. When this happens, the data from a previously closed frame (a frame whose closing flag has been received) will not be destroyed.
5. **Servicing the Rx FIFO in a 2-Byte Mode** — The procedure for reading the last bytes of data is the same, regardless of whether the frame contains an even or an odd number of bytes. Continue to read 2 bytes until an interrupt cocurs that is caused by an end-of-frame status (FV or ERR) When this occurs, indicating the last byte either has been read or is ready to be read, switch temporarily to the 1-byte mode with no prioritized status (control register 2) Test RDA to indicate whether a 1-byte read should be performed. Then clear the frame end status.
6. **Frame Complete Status and $\overline{RTS}$ Release** — In many cases, a MODEM will require a delay for releasing $\overline{RTS}$. An 8-bit or 16-bit delay can be added to the ADLC $\overline{RTS}$ output by using an Abort. At the end of a transmission, frame complete status will indicate the frame completion. After frame complete status goes high, write "1" into the Abt control bit (and Abt Extend bit if a 16-bit delay is required). After the Abt control bit is set, write "0" into the $\overline{RTS}$ control bit. The transmitter will transmit eight or sixteen "1's" and the $\overline{RTS}$ output will then go high (inactive).
7. **Note to users not using the MC6800** — (a) Care should be taken when performing a write followed by a read on successive E pulses at a high frequency rate. Time must be allowed for status changes to occur. If this is done, the time that E is low between successive write/read E pulses should be at least 500 ns. (b) The ADLC is a completely static part. However, the E frequency should be high enough to move data through the FIFOs and to service the peripheral requirements. Also, the period between successive E pulses should be less than the period of RxC or TxC in order to maintain synchronization between the data bus and the peripherals
8. **Clear-to-Send ($\overline{CTS}$)** — The $\overline{CTS}$ input, when high, provides a real-time inhibit to the TDRA status bit and its associated interrupt. All other status bits will be operational. Since it inhibits TDRA, $\overline{CTS}$ also inhibts the TDSR DMA request. The $\overline{CTS}$ input being high does not affect any other part of the transmitter Information in the Tx FIFO and Tx Shift Register will, therefore, continue to be transmitted as long as the Tx CLK is running.

MECHANICAL DATA

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Cerdip S Suffix	1.0	0°C to 70°C	MC6854S
	1.0	–40°C to 85°C	MC6854CS
	1.5	0°C to 70°C	MC68A54S
	1.5	–40°C to 85°C	MC68A54CS
	2 0	0°C to 70°C	MC68B54S
Plastic P Suffix	1.0	0°C to 70°C	MC6854P
	1.0	–40°C to 85°C	MC6854CP
	1.5	0°C to 70°C	MC68A54P
	1.5	–40°C to 85°C	MC68A54CP

PIN ASSIGNMENT


V_{SS} 1
$\overline{\text{RTS}}$ 2
RxD 3
RxC 4
TxC 5
TxD 6
$\overline{\text{IRQ}}$ 7
$\overline{\text{RESET}}$ 8
$\overline{\text{CS}}$ 9
RS0 10
RS1 11
R/$\overline{\text{W}}$ 12
E 13
V_{CC} 14
28 $\overline{\text{CTS}}$
27 $\overline{\text{DCD}}$
26 $\overline{\text{LOC}}$/$\overline{\text{DTR}}$
25 $\overline{\text{FLAG DET}}$
24 TDSR
23 RDSR
22 D0
21 D1
20 D2
19 D3
18 D4
17 D5
16 D6
15 D7

MC68488

General Purpose Interface Adapter

The MC68488 GPIA provides the means to interface between the IEEE-488 standard instrument bus and the M6800 MPU Family. The GPIB instrument bus provides a means of controlling and moving data between instruments connected to it.

The MC68488 will automatically handle all handshake protocol needed on the instrument bus.

- Single- or Dual-Primary Address Recognition
- Secondary Address Capability (Talker or Listener)
- Complete Source and Acceptor Handshakes
- Programmable Interrupts
- RFD Holdoff to Prevent Data Overrun
- Operates with DMA Controller
- Serial- and Parallel-Polling Capability
- Talk-Only or Listen-Only Capability
- Selectable Automatic Features to Minimize Software
- Synchronization Trigger Output
- M6800 Bus Compatible

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

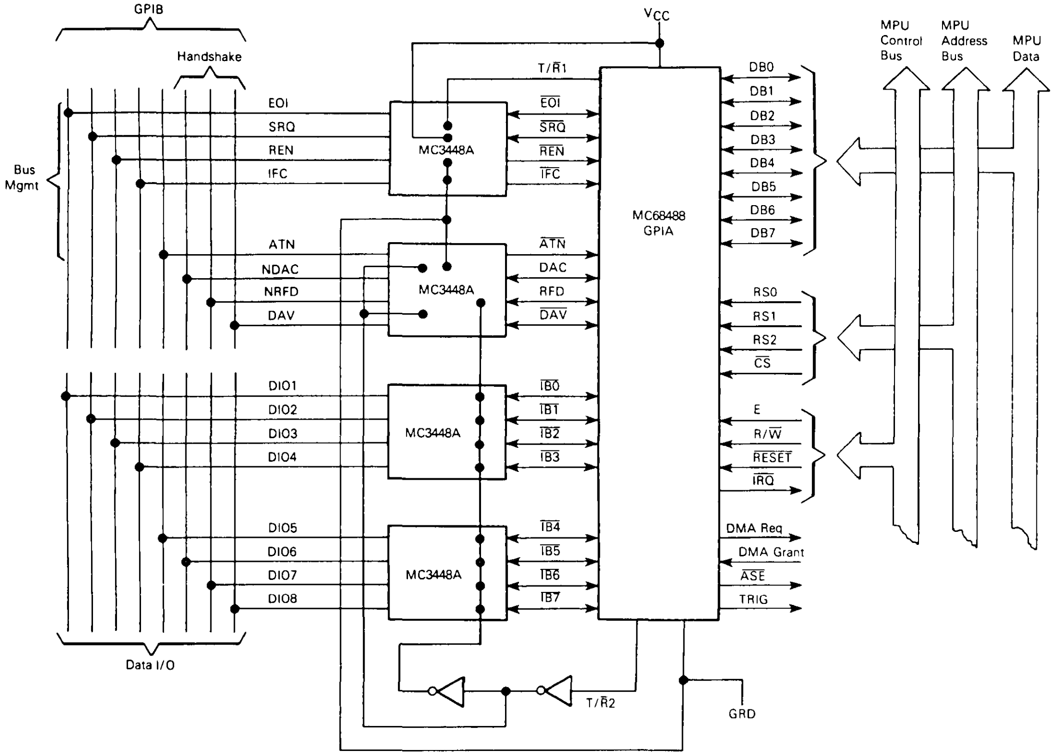
THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance	θ_{JA}		°C/W
Cerdip		60	
Plastic		100	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — GPIB INTERFACE

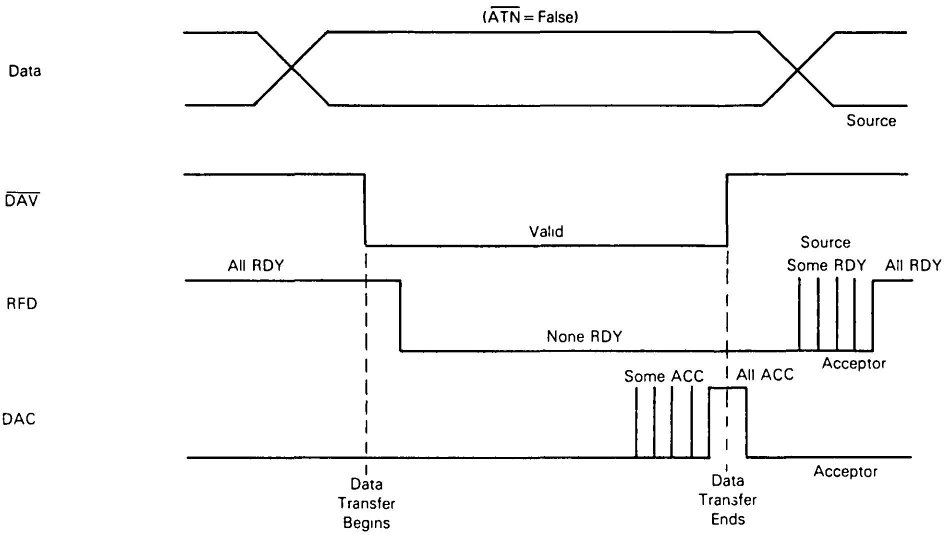

GPIB
Handshake
Bus Mgmt
EOI
SRQ
REN
IFC
ATN
NDAC
NRFD
DAV
DIO1
DIO2
DIO3
DIO4
DIO5
DIO6
DIO7
DIO8
Data I/O
V_{CC}
T/$\overline{R}$1
MC3448A
MC3448A
MC3448A
MC3448A
$\overline{EOI}$
$\overline{SRQ}$
$\overline{REN}$
$\overline{IFC}$
$\overline{ATN}$
DAC
RFD
$\overline{DAV}$
$\overline{IB0}$
$\overline{IB1}$
$\overline{IB2}$
$\overline{IB3}$
$\overline{IB4}$
$\overline{IB5}$
$\overline{IB6}$
$\overline{IB7}$
MC68488 GPIA
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
RS0
RS1
RS2
$\overline{CS}$
E
R/$\overline{W}$
$\overline{RESET}$
$\overline{IRQ}$
DMA Req
DMA Grant
$\overline{ASE}$
TRIG
T/$\overline{R}$2
GRD
MPU Control Bus
MPU Address Bus
MPU Data

Note The four MC3448A quad bus transceivers can be replaced by two MC3447 octal bus transceivers

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5 0 Vdc ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2 0	–	V_{CC}	V
Input Low Voltage		V_{IL}	V_{SS} − 0 3	–	V_{SS} + 0 8	V
Input Leakage Current (V_{in} = 0 to 5 25 V)		I_{in}	–	1 0	2 5	μA
Three State (Off State) Input Current (V_{in} = 0 4 to 2 4 V)	D0-D7	I_{TSI}	–	2 0	10	μA
DC Output High Voltage (I_{load} = −205 μA)	D0-D7	V_{OH}	V_{SS} + 2 4	–	–	V
DC Output Low Voltage (I_{Load} = 1 6 mA) (I_{Load} = 3 2 mA)	 D0-D7 $\overline{SRQ}$, $\overline{IRQ}$	V_{OL}	 – –	 – –	 V_{SS} + 0 4 V_{SS} + 0 4	V
Output Leakage Current (Off State) (V_{OH} = 2 4 V)	$\overline{SRQ}$, $\overline{IRQ}$	I_{LOH}	–	1 0	10	μA
Internal Power Dissipation		P_{INT}	–	600	750	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1 0 MHz)	 D0-D7 All Others	C_{in}	 – –	 – –	 12 5 7 5	pF

FIGURE 2 — SOURCE AND ACCEPTOR HANDSHAKE


($\overline{ATN}$ = False)
Data
Source
$\overline{DAV}$
Valid
All RDY
Source
Some RDY
All RDY
RFD
None RDY
Acceptor
Some ACC
All ACC
DAC
Acceptor
Data Transfer Begins
Data Transfer Ends

This diagram displays logical voltage levels on the MC68488 pins The MC68488 pins are labeled as the complement of the specified 488 bus callout, i e , $\overline{DAV}$ rather than DAV RFD rather than NRFD and DAC rather than NDAC This was done to stay with standard positive logic format, which is used with all M6800 family devices

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from.

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad (1)$$

Where·

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \qquad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

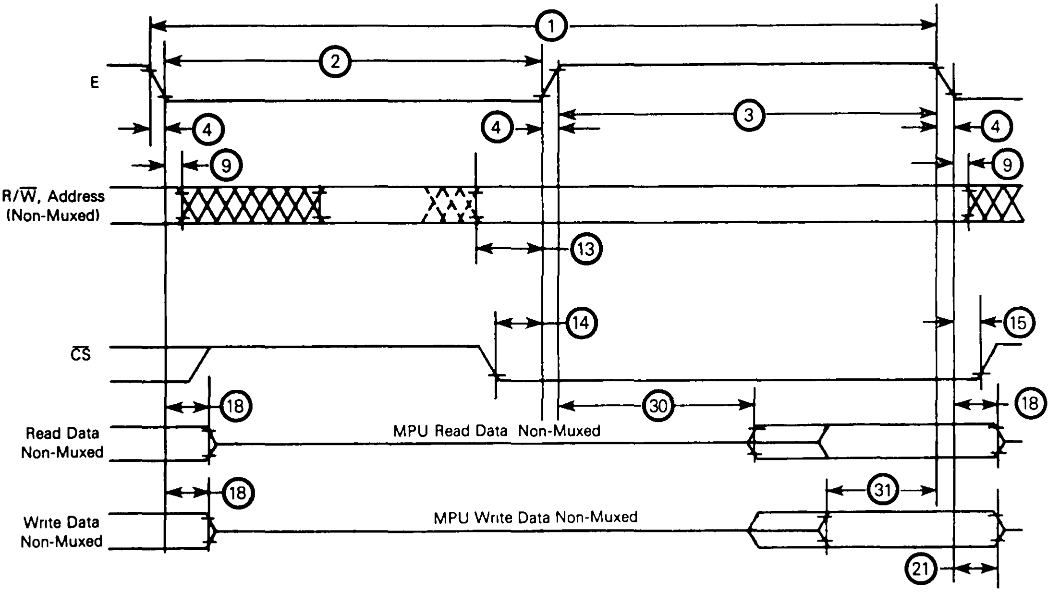
BUS TIMING (See Notes 1, 2, and 3)

Ident. Number	Characteristics	Symbol	MC68488		MC68A488*		MC68B488*		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50**	20	50**	20	50**	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*See Table 1 for GPIB transceiver considerations when using MC68A488 or MC68B488

**The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high-impedance)

FIGURE 3 — BUS TIMING


E
R/W̄, Address (Non-Muxed)
C̄S̄
Read Data Non-Muxed
Write Data Non-Muxed
MPU Read Data Non-Muxed
MPU Write Data Non-Muxed
①
②
③
④
⑨
⑬
⑭
⑮
⑱
㉑
㉚
㉛

NOTES
1 Not all signals are applicable to every part
2 Voltage levels shown are $V_L \leq 0.8$ V, $V_H \geq 2.4$ V, unless otherwise specified
3 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FIGURE 4 — OUTPUT BUS TIMING

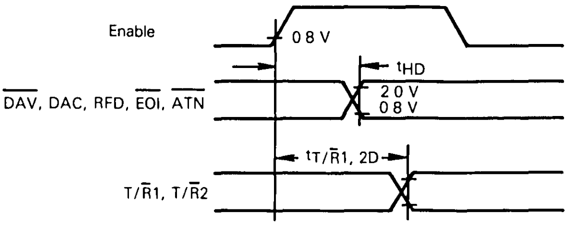


TABLE 1 — AC TIME VALUES

Characteristics		Symbol*	Typ	Unit
Settling Time for Multiple Message	SH	T_1	≥2	μs**
Response to $\overline{ATN}$	SH, AH, T, L	t_3	≤200	ns
Interface Message Accept Time‡	AH	T_3	>0	§
Response to $\overline{IFC}$ or $\overline{REN}$ False	T, TE, L, LE	t_4	<100	μs
Response to $\overline{ATN}\bullet\overline{EOI}$	PP	t_5	≤200	ns

* Time values specified by a lower case t indicate the maximum time allowed to make a state transition Time values specified by an upper case T indicate the minimum time that a funcion must remain in a state before exiting

** If three-state drivers are used on the DIO-DAV and EOI lines, T_1 may be
(1) ≥ 1100 ns
(2) Or ≥700 ns if it is known that within the controller ATN is driven by a three-state driver
(3) Or ≥500 ns for all subsequent bytes following the first sent after each false transition of ATN [the first byte must be sent in accordance with (1) or (2)]

‡ Time required for interface functions to accept, not necessarily respond to interface messages.

§ Implementation dependent

When using an E clock of 1 5 MHz on the MC68A488, the GPIB data lines, DAV, and EOI lines must have three-state drivers – See Note **
When using an E clock of 2 0 MHz on the MC68B488 the GPIB data lines, DAV, EOI, and ATN lines must have three-state drivers – See Note **

GENERAL DESCRIPTION

The IEEE-488 instrument bus standard is a bit-parallel, byte-serial bus structure designed for communication to and from intelligent instruments Using this standard, many instruments may be interconnected, remotely and automatically controlled, or programmed Data may be taken from, sent to, or transferred between instruments A bus controller dictates the role of each device by making the attention line true and sending talk or listen addresses on the instrument bus data lines, those devices which have matching addresses are activated Device addresses are set into each GPIA from switches or jumpers on a PC board by a microprocessor as a part of the initialization sequence

When the controller makes the attention line true, instrument bus commands may also be sent to single or multiple GPIAs

Information is transmitted on the instrument bus data lines under sequential control of the three handshake lines No step in the sequence can be initiated until the previous step is completed Information transfer can proceed as fast as the devices can respond, but no faster than the slowest device presently addressed as active This permits several devices of different speeds to receive the same data concurrently.

The GPIA is designed to work with standard 488-bus driver ICs (MC3447As or MC3448As) to meet the complete electrical specifications of the IEEE-488 bus Additionally, a powered-off instrument may be powered-on without disturbing the 488 bus With some additional logic, the GPIA could be used with other microprocessors

The MC68488 GPIA has been designed to interface between the M6800 family microprocessor and the complex protocol of the IEEE-488 instrument bus Many instrument bus protocol functions are handled automatically by the GPIA and require no additional MPU action Other functions require minimum MPU response due to a large number of internal registers conveying information on the state of the GPIA and the instrument bus

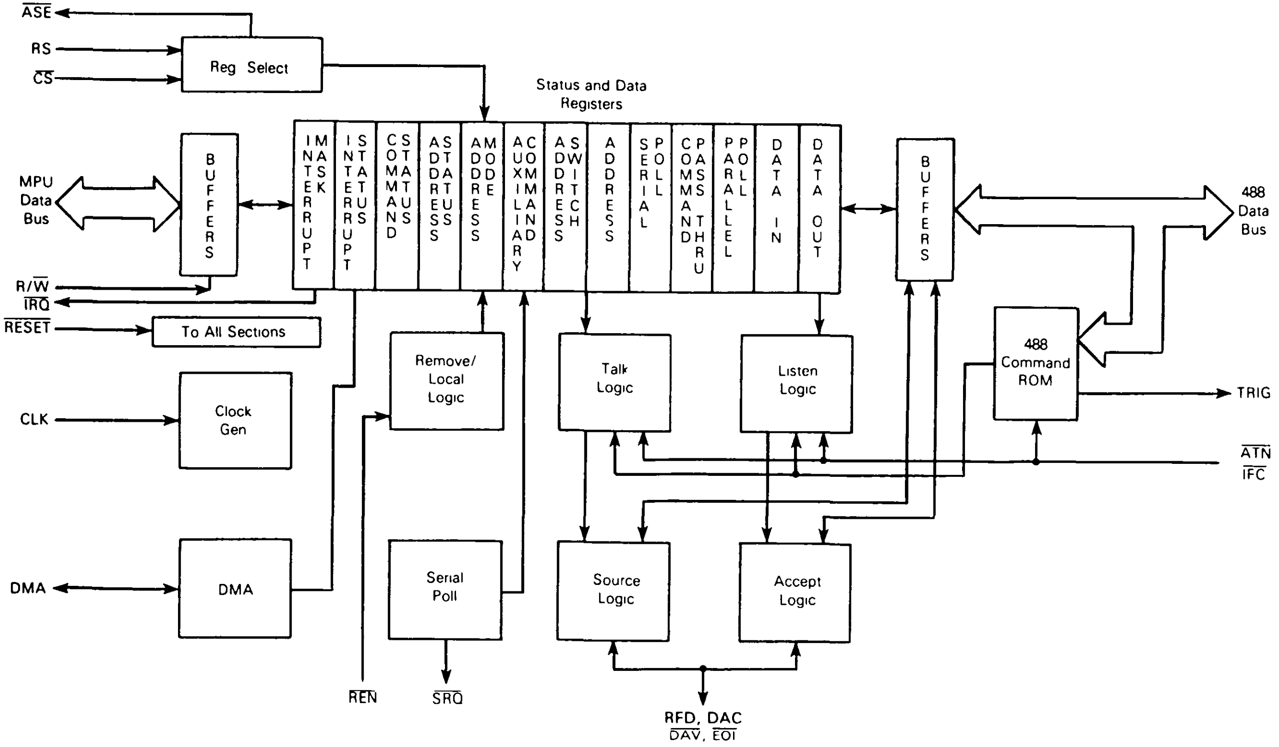
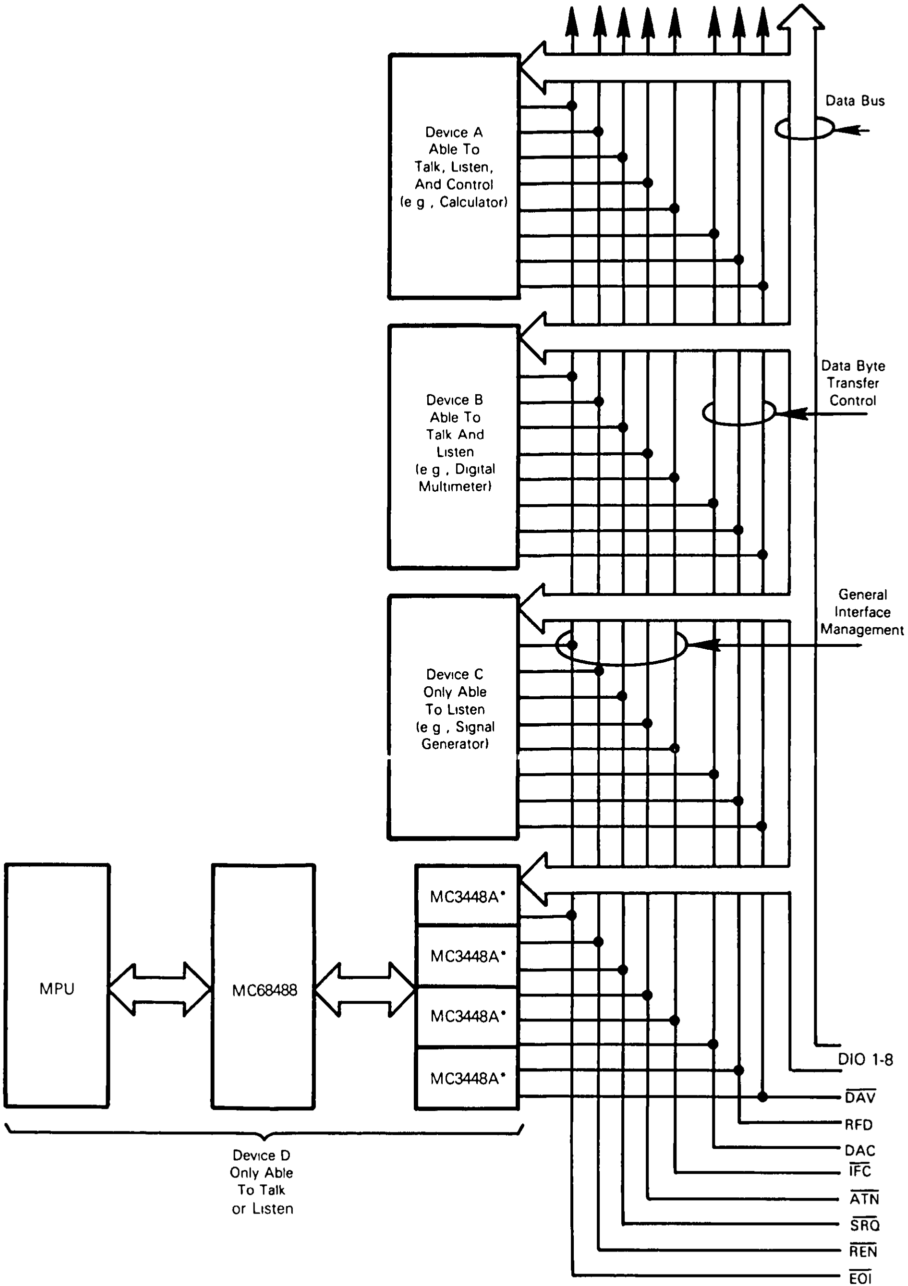

ASE
RS
CS
Reg Select
Status and Data Registers
MPU Data Bus
BUFFERS
INTERRUPT MASK
INTERRUPT STATUS
COMMAND STATUS
ADDRESS STATUS
ADDRESS MODE
AUXILIARY COMMAND
ADDRESS SWITCH
ADDRESS
SERIAL POLL
COMMAND PASS THRU
PARALLEL POLL
DATA IN
DATA OUT
BUFFERS
488 Data Bus
R/W
IRQ
RESET
To All Sections
Remove/ Local Logic
Talk Logic
Listen Logic
488 Command ROM
TRIG
ATN
IFC
CLK
Clock Gen
DMA
DMA
Serial Poll
Source Logic
Accept Logic
REN
SRQ
RFD, DAC
DAV, EOI

FIGURE 5 — GPIA BLOCK DIAGRAM

FIGURE 6 — GPIB SYSTEM


Data Bus
Device A
Able To
Talk, Listen,
And Control
(e g , Calculator)
Data Byte
Transfer
Control
Device B
Able To
Talk And
Listen
(e g , Digital
Multimeter)
General
Interface
Management
Device C
Only Able
To Listen
(e g , Signal
Generator)
MPU
MC68488
MC3448A*
MC3448A*
MC3448A*
MC3448A*
DIO 1-8
DAV
RFD
DAC
IFC
ATN
SRQ
REN
EOI
Device D
Only Able
To Talk
or Listen

*Two MC3447A octal transceivers can be used in place of four MC3448As

PIN DESCRIPTION

All inputs to the GPIA are high impedance and TTL compatible All outputs from the GPIA are compatible with standard TTL IRQ (Interrupt Request) and SRQ, however, are open-drain outputs (no internal pullup)

INTERFACE WITH MPU

BIDIRECTIONAL DATA (D0-D7) – The bidirectional data lines allow the transfer of data between the MPU and GPIA The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a GPIA read operation or the DMA controller performs a memory write operation The Read/Write line is high when the GPIA is selected for a read operation

CHIP SELECT (CS) – This input signal is used to select the GPIA. CS must be low for selection of the device Chip Select decoding is normally accomplished with external logic

READ/WRITE INPUT (R/W) – This signal is generated by the MPU or DMA controller to control register access and direction of data transfer on the data bus A low state on the GPIA Read/Write and DMA Grant lines allows for the selection of one of seven write-only reigsters when used in conjunction with register select lines RS0, RS1, and RS2 A high state on the GPIA Read/Write and low state on the DMA Grant line allows for the selection of one of eight read-only registers when used in conjunction with register select lines RS0, RS1, and RS2.

REGISTER SELECT (RS0, RS1, RS2) – The three register select inputs are used to select the various registers inside the GPIA These three lines are used in conjunction with the Read/Write line to select a particular register that is to be written or read Table 2 shows the register select coding

INTERRUPT REQUEST (IRQ) – The IRQ output goes to the common interrupt bus line for the MPU This is an open-drain output which is wire-ORed to the IRQ bus line The IRQ is asserted low when an enable interrupt occurs and stays low until the MPU reads the interrupt status register Reading R0R will reset IRQ to the high state

TABLE 2 – REGISTER ACCESS

RS2	RS1	RS0	R/W	Register Title	Register Symbol
0	0	0	1	Interrupt Status	R0R
0	0	0	0	Interrupt Mask	R0W
0	0	1	1	Command Status	R1R
0	0	1	0	Unused	–
0	1	0	1	Address Status	R2R
0	1	0	0	Address Mode	R2W
0	1	1	1	Auxiliary Command	R3R
0	1	1	0	Auxiliary Command	R3W
1	0	0	1	Address Switch*	R4R
1	0	0	0	Address	R4W
1	0	1	1	Serial Poll	R5R
1	0	1	0	Serial Poll	R5W
1	1	0	1	Command Pass-Through	R6R
1	1	0	0	Parallel Poll	R6W
1	1	1	1	Data In	R7R
1	1	1	0	Data Out	R7W

*External to MC68488

RESET — The RESET input provides a means of resetting the GPIA from a hardware source. In the low state, the RESET input causes the following:

- The Interrupt "Mask" register is reset;
- All status conditions are reset;
- The GPIA is placed in the Untalk/Unlisten state;
- The Parallel Poll, Serial Poll, Data In, and Data Out registers are reset;
- The Address register and Address mode register are cleared;
- All stored conditions in the Auxiliary Command register except bit 7 are reset — (bit 7 is set);
- T/R1, 2 will go to the low state.

When RESET returns high (the inactive state) the GPIA will remain in the reset condition until the MPU writes bit 7 of the Auxiliary Command register (R3W) low Prior to the release of the software reset bit, the only register that can be accessed is the Address register The conditions affected by the RESET pin cannot be changed while this pin is low

E (ENABLE CLOCK) – E activates the address inputs (CS, RS0, RS1, and RS2) and R/W input and enables data transfer on the MPU data bus It is also used internally as a state counter allowing the device to change interface states The E input should be connected to a free-running clock source such as the MC6800 $\phi 2$ or the Enable Signal of other M6800 family MPUs

GPIA/GPIB INTERFACE BUS SIGNALS

The GPIA provides a set of eighteen interface signal lines between the M6800 and the IEEE-488 Standard bus

NOTE

> The IEEE-488 Standard defines these signals as negative logic In this document all MPU and MC68488 signals are defined as positive logic

SIGNAL LINES (IB0-IB7) – These bidirectional lines allow for the flow of 7-bit ASCII interface messages and device dependent messages Data appears on these lines in a bit-parallel byte-serial form These lines are buffered by transceivers and applied to the IEEE-488 Standard bus (DIO1-DIO8)

BYTE TRANSFER LINES (DAC, RFD, DAV) – These lines allow for proper transfer of each data byte on the bus between sources and acceptors RFD goes passively high indicating "Ready For Data" A source will indicate the "data is valid" by pulling DAV low Upon the reception of valid data, DAC will go passively high indicating that the "data has been accepted" by all acceptors The handshake lines have internal pullup resistors

BUS MANAGEMENT LINES (ATN, IFC, SRQ, EOI, REN) – These lines are used to manage an orderly flow of information across the interface lines

ATTENTION (ATN) – is continuously monitored by the GPIA The device responds to any changes on this line in less than 200 ns by activating the transmit/receive control signals If the EOI line and ATN are low at the same time, GPIA will place the contents of a parallel poll register on the IEEE-488 Standard bus

INTERFACE CLEAR (IFC) — is used by a system controller to put the GPIA in a known quiescent state. The occurrence of IFC will place the GPIA in the Listener/Talker idle state (LIDS or TIDS) If the MC68488 is in a Listener Active state with a byte of data in the Data-In register (BI bit set) an IFC will place the part in LIDS but will not destroy the received byte nor the status indication (BI). Any interface function that requires the device to be in either the Listener or Talker Active state (e g , a Serial Poll enable command) will be reset if an IFC occurs. A command that originates from the MPU (e g , to, lo, fget, hlda) will only be affected during the occurrence of an IFC (when IFC is low) and will return to its programmed state when IFC returns high, i.e., IFC will not affect local messages For example if the GPIA is in TACS (Talker Active State) and has placed a byte in the Data-Out register it has made a new byte available (nba) If IFC occurs while the source handshake is in SDYS, the talker function will be returned to its idle state but nba (a local message) will not be destroyed When the GPIA is again made a talker, the byte in the Data-Out register (placed there before IFC) will be placed onto the GPIB The address register is not affected by an IFC

SERVICE REQUEST (SRQ) — is used to indicate a need for attention in addition to requesting an interruption in the current sequence of events This indicates to the controller that a device on the bus is in need of service This output becomes active low by setting the rsv bit (bit 6) of R5W This line is an open drain and an external pullup resistor (nominal 3 3k ohm) must be used

REMOTE ENABLE (REN) — is used to select one of two alternate sources of device programming data — local and remote control When this input is low the GPIA is enabled to move to the REMS state Note that REN being low is a necessary but not a sufficient condition for moving to REMS

END OF IDENTIFY (EOI) — Serves a dual purpose When the GPIA is in TACS and the MPU writes bit 5 or R3W (feoi) this pin becomes an output and signals the end of a multibyte transfer If the system controller makes the EOI line true in conjunction with ATN, the contents of the Parallel Poll register will be placed on the IEEE-488 Standard bus

TRANSMIT/RECEIVE CONTROL SIGNALS (T/R̄1, T/R̄2) — These two signals are used to control the quad or octal transceivers which drive the interface bus It is assumed that transceivers equivalent to the MC3447 or MC3448A will be used where each transceiver has a separate Transmit/Receive control pin These pins can support one TTL load each The outputs can then be grouped and the control for SRQ hardwired high to transmit The Transmit/Receive inputs of REN, IFC, and ATN are hardwired low to receive EOI is controlled by T/R̄1 through the MC3447/MC3448A (or equivalents) allowing it to transmit or receive T/R̄1 operates exactly as T/R̄2 except during the parallel polling sequence During parallel poll, EOI will be made an input by T/R̄1 while DAV and IB0/IB7 lines are outputs

SPECIAL CONTROL SIGNALS

DMA CONTROL LINES (DMA GRANT, DMA REQUEST — The DMA request line is used to signal a DMA controller that a data transfer is pending. The DMA request line is set high if either the BI or BO status bits are set in the Interrupt Status Register (R0R). The DMA request line is cleared when the DMA Grant is true The DMA Grant line is used to signal the GPIA that the DMA has control of the MPU data and address lines The DMA Grant, when set high, selects register 7. It also inhibits the RS0, RS1, and RS2 lines. During this time the CS input must be high. The DMA Grant also inverts the function of the R/W line making it R̄/W. Thus, if the DMAC supplies a write function to a memory location, this same line will perform a read of the GPIA (R7R) and vice versa.

NOTE

DMA GRANT MUST BE GROUNDED WHEN NOT IN USE

TRIGGER OUTPUT (TRIG) — The TRIG pin provides an output corresponding to the GET and fget commands. A hardware or software reset places this output at a low level. The trigger output can be programmed high by either of two methods:

1. Setting fget (bit 0 of R3W) by the MPU causes the trigger output to be set. It remains set until the fget bit is programmed low or until a reset occurs
2. The Trigger Output is set upon reception of a GET command from the controller It is reset when the GPIA moves out of DTAS (Device Trigger Active State), i e., when GET, LADS, or ACDS occur

ADDRESS SWITCH ENABLE (ASE) — The ASE output is used to enable three-state buffers that connect instrument address switches to the MPU data bus This output pin is pulsed low when the Address Switch Register of the GPIA is read (R4R), i.e , a read of R4R will drive the ASE line low for the E clock that is used to read R4R.

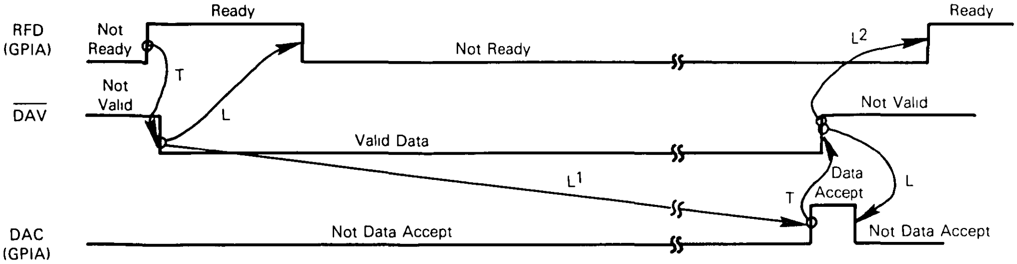
GPIB HANDSHAKE SEQUENCE

The GPIB handshake line transitions are debounced inside the GPIA with the E-clock to provide a high degree of noise immunity Due to the asynchronous nature between the GPIB handshake line transitions and the internal debounce circuit sampling, the time required for handshake completion can vary by 1 E clock cycle.

LISTENER MODE — The handshake sequence begins when the GPIA makes RFD true (Figure 8). A second byte cannot be transferred on the GPIB until the GPIA again makes RFD true for the next handshake The total time required by the GPIA to debounce all of the handshake lines in the appropriate time sequence is 7-8 E clock cycles The 1 cycle variation is due to the asynchronous nature of the GPIB with respect to the GPIA debounce circuitry. To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer.

TALKER MODE – The handshake sequence begins when the listener(s) on the GPIB make the RFD line true (Figure 9). When this occurs and the MPU has written a byte to R7W the GPIA will make the $\overline{DAV}$ line low indicating to the listeners that valid data is on the GPIB When this byte is accepted and RFD is again made true the next transfer can begin The GPIA debounce circuitry requires 6-7 E clock cycles to complete a handshake sequence. As with the listener there is a 1 E clock fluctuation due to the asynchronous nature between the GPIB handshake and the GPIA debounce circuitry To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer

FIGURE 7 — GPIA IN LISTEN MODE*


RFD
(GPIA)
DAV
DAC
(GPIA)
Ready
Not Ready
Not Ready
Ready
Not Valid
T
L
Valid Data
L2
Not Valid
L1
Data Accept
T
L
Not Data Accept
Not Data Accept

- The GPIA in the listener mode controls the DAC and RFD lines The $\overline{DAV}$ line is controlled by the Talker on the GPIB Note that the RFD and DAC lines are wire ANDed on the GPIB, thus, these lines returning to the high state are dependent on all devices programmed as listeners releasing the lines

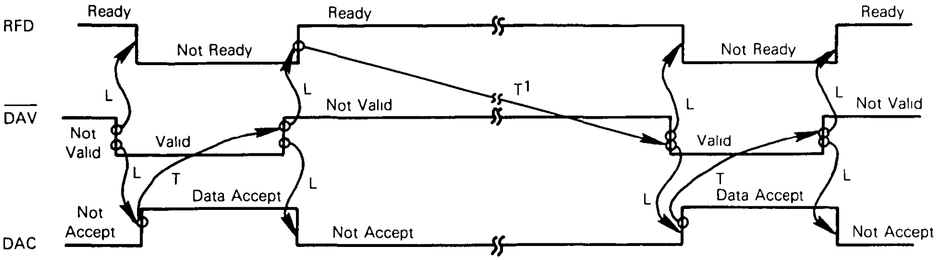
L The listener(s) on the GPIB causes this action

T The talker on the GPIB causes this action

1 The release of DAC may require action by the MPU (reading R7R for data byte transfers or writing dacr (R3W) high for certain commands) For some commands DAC is automatically released by the GPIA Consult the MC68488 user's manual for details

2 The RFD line is normally automatically released by the GPIA Certain conditions, however, require MPU intervention to provide this release Consult the MC68488 user's manual for details

FIGURE 8 — GPIA IN TALKER MODE*


RFD
DAV
DAC
Ready
Not Ready
Ready
Not Ready
Ready
L
L
T1
L
L
Not Valid
Not Valid
Not Valid
Valid
Valid
L
T
L
L
T
L
Data Accept
Data Accept
Not Accept
Not Accept
Not Accept

- The GPIA in the talker mode controls the $\overline{DAV}$ line The RFD and DAC lines are controlled by the listener(s) on the GPIB

L The listener(s) on the GPIB causes this action

T The talker on the GPIB causes this action

1 Two conditions must occur before the $\overline{DAV}$ line goes to the valid state The RFD line must be high and a data byte must be placed in the data out register (nba must be true)

GPIA INTERNAL CONTROLS AND REGISTERS*

There are fifteen locations accessible to the MPU data bus which are used for transferring data to control the various functions on the chip and provide current chip status Seven of these registers are write only and eight registers are read only. The various registers are accessed according to the three least-significant bits of the MPU address bus and the status of the Read/Write line. One of the fifteen registers is external to the IC but an address switch register is provided for reading the address switches Table 2 shows actual bit contents of each of the registers.

DATA-IN REGISTER R7R — The data-in register is an actual 8-bit storage register used to move data from the interface bus when the chip is a listener. Reading the register does not destroy information in the data-out register DAC (data accepted) will remain low until the MPU removes the bytes from the data-in register The chip will automatically finish the handshake by allowing DAC to go high. In RFD (ready for data) holdoff mode, a new handshake is not initiated until a command is sent allowing the chip to release holdoff. This will delay a talker until the available information has been processed

Data-In Register
(Read Only)

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

DI0-DI7 — Correspond to DIO1-DIO8 of the 488-1975 Standard and $\overline{\text{IB0}}$-$\overline{\text{IB7}}$ of the MC68488

*NOTE Upper and lower case type designations will be used with the register bits to indicate remote or local messages respectively

DATA-OUT REGISTER R7W — The data-out register is an actual 8-bit storage register used to move data out of the chip onto the interface bus. Reading from the data-in register has no effect on the information in the data-out register. Writing to the data-out register has no effect on the information in the data-in register.

Data-Out Register
(Write Only)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

DO0-DO7 — Correspond to DIO1-DIO8 of the 488-1978 Standard and $\overline{\text{IB0}}$-$\overline{\text{IB7}}$ of the MC68488

INTERRUPT MASK REGISTER R0W — The Interrupt Mask Register is a 7-bit storage register used to select the particular events that will cause an interrupt to be sent to the MPU The seven control bits may be set independently of each other. If dsel (bit 7 of the Address Mode Register) is set high CMD bit 2 will interrupt on SPAS or RLC. If dsel is set low CMD will interrupt on UACG, UUCG, and DCAS in addition to RLC and SPAS. The Command Status Register R1R may then be used to determine which command caused the interrupt. Setting GET bit 5 allows an interrupt to occur on Group Execute Trigger Command. END bit 1 allows an interrupt to occur if $\overline{\text{EOI}}$ is true (low) and $\overline{\text{ATN}}$ is false (high). APT bit 3 allows an interrupt to occur indicating that a secondary address is available to be examined by the MPU if apte (bit 0 of Address Mode Register) is enabled and listener or talker primary address is received and a Secondary Command Group is received. A typical response for a valid secondary address would be to set msa (bit 3 of Auxiliary Command Register) true and dacr (bit 4 Auxiliary Command Register) true, releasing the DAC handshake BI indicates that a data

TABLE 3 — REGISTER CONTENTS

	7	6	5	4	3	2	1	0	
R0W	IRQ	BO	GET		APT	CMD	END	BI	Interrupt "Mask Register"
R0R	INT	BO	GET		APT	CMD	END	BI	Interrupt Status Register
R1R	UACG	REM	LOK		RLC	SPAS	DCAS	UUCG	Command Status Register
R1W									Unused
R2R	ma	to	lo	ATN	TACS	LACS	LPAS	TPAS	Address Status Register
R2W	dsel	to	lo		hlde	hlda		apte	Address Mode Register
R3R	RESET	DAC	DAV	RFD	msa	rtl	ulpa	fget	Auxiliary Command Register
R3W		rfdr	feoi	dacr			dacd		Auxiliary Command Register
R4R	UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1	Address Switch Register
R4W	lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1	Address Register
R5R	S7	SRQS	S5	S4	S3	S2	S1	S0	Serial Poll Register
R5W		rsv							
R6R	B7	B6	B5	B4	B3	B2	B1	B0	Command Pass-Through Register
R6W	PPR8	PPR7	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	Parallel Poll Register
R7R	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Data In Register
R7W	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	Data Out Register

Notes

1 Upper case letters indicate a message resulting from the IEEE-488 Standard bus
2 Lower case letters indicate a message resulting from the MPU data bus
3 The bit terminology of the Data In and Data registers represent the numbering of the IEEE-488 Standard bus and not the 6800 MPU bus — see Section 3 1 2

byte is waiting in the data-in register. BI is set high when data-in register is full. BO indicates that a byte from the data-out register has been accepted. BO is set when the data-out register is empty. IRQ enabled high allows any interrupt to be passed to the MPU.

Interrupt Mask Register
(Write Only)

IRQ	BO	GET	X	APT	CMD	END	BI

IRQ – Mask bit for IRQ pin
BO – Interrupt on byte output
GET – Interrupt on Group Execute Trigger
APT – Interrupt on Secondary Address Pass-Through
CMD – Interrupt on SPAS + RLC + $\overline{\text{dsel}}$ (DCAS + UUCG + UACG)
END – Interrupt on EOI and $\overline{\text{ATN}}$
BI – Interrupt on byte input

THE INTERRUPT STATUS REGISTER R0R – The Interrupt Status Register is a 7-bit storage register which corresponds to the interrupt mask register with an additional bit INT bit 7. Except for the INT bit the other bits in the status register are set regardless of the state of the interrupt mask register when the corresponding event occurs. The $\overline{\text{IRQ}}$ (MPU interrupt) is cleared when the MPU reads from the register. INT bit 7 is the logical OR of the other six bits ANDed with the respective bit of R0W.

Interrupt Status Register
(Read Only)

INT	BO	GET	X	APT	CMD	END	BI

INT – Logical OR of all other bits in this register ANDed with the respective bits in the interrupt mask register
BO – A byte of data has been output
GET – A Group Execute Trigger has occurred
APT – An Address Pass-Through has occurred
CMD – SPAS + RLC + $\overline{\text{dsel}}$ (DCAS + UUCG + UACG) has occurred
END – An EOI has occurred with $\overline{\text{ATN}}$ = 0
BI – A byte has been received

SERIAL POLL REGISTER R4R/W – The Serial Poll Register is an 8-bit storage register which can be both written into and read by the MPU. It is used for establishing the status byte that the chip sends out when it is serial poll enabled. Status may be placed in bits 0 through 5 and bit 7. Bit 6 rsv (request for service) is used to drive the logic which controls the $\overline{\text{SRQ}}$ line on the bus telling the controller that service is needed. This same logic generated the signal SRQS which is substituted in bit 6 position when the status byte is read by the MPU $\overline{\text{IB0-IB7}}$. In order to initiate a rsv (request for service), the MPU sets bit 6 true (generating $\overline{\text{rsv}}$ signal) and this in turn causes the chip to pull down the $\overline{\text{SRQ}}$ line. SRQS is the same as rsv when SPAS is false. Bit 6 as read by the MPU will be the SRQS (Service Request State).

Serial Poll Register
(Read)

S8	SRQS	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
SRQS – Bus in Service Request State

Serial Poll Register
(Write)

S8	rsv	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
rsv – generate a service request

PARALLEL POLL REGISTER R6W – This register will be loaded by the MPU and the bits in this register will be delivered to the instrument bus IB0-IB7 during PPAS (Parallel Poll Active State). This register powers up in the PPO (Parallel Poll No Capability) state. The reset bit (Auxiliary Command Register bit 7) will clear this register to the PPO state.

The parallel poll interface function is executed by this chip using the PP2 subset (Omit Controller Configuration Capability). The controller cannot directly configure the parallel poll output of this chip. This must be done by the MPU. The controller will be able to indirectly configure the parallel poll by issuing an addressed command which has been defined in the MPU software.

Parallel Poll Register
(Write Only)

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bits delivered to bus during Parallel-Poll Active State (PPAS)
Register powers up in the PP0 state
Parallel Poll is executed using the PP2 subset

ADDRESS MODE REGISTER R2W – The address mode register is a storage register with six bits for control to, lo, hlde, hlda, dsel, and apte. The to bit 6 selects the talker/listener and addresses the chip to talk only. The lo bit 5 selects the talker/listener and sets the chip to listen only. The apte bit 0 is used to enable the extended addressing mode. If apte is set low the device goes from the TPAS (Talker Primary Address State) directly to the TADS (Talker Addressed State). The hlda bit 2 holds off RFD (Ready for Data) on ALL DATA until rfdr is set true. The hlde bit 3 holds off RFD on EOI enabled (low) and ATN not enabled (high). This allows the last byte in a block of data to be continually read as needed. Writing rfdr true (high) will allow the next handshake to proceed.

Address Mode Register
(Write Only)

dsel	to	lo	X	hdle	hdla	X	apte

dsel – configure for automatic completion of handshake sequence on occurrence of GET, UACG, UUCG, SDC, or DCL commands
to – set to talk-only mode
lo – set to listen-only mode
hdle – Hold-off RFD on end (END = EOI∧$\overline{\text{ATN}}$)
hdla – Hold-off RFD on all data
apte – Enable the address pass-through feature

ADDRESS STATUS REGISTER R2R – The address status register is not a storage register but simply an 8-bit port used to couple internal signal modes to the MPU bus The status flags represented here are stored internally in the logic of the chip. These status bits indicate the addressed state of the talker/listener as well as flags that specify whether the chip is in the talk only or listen only mode. The ATN, bit 4, contains the condition of the Attention Line The ma signal is true when the chip is in.

TACS – Talker Active State
TADS – Talker Addressed State
LACS – Listener Active State
LADS – Listener Addressed State
SPAS – Serial Poll Active State

Address Status Register
(Read Only)

ma	to	lo	ATN	TACS	LACS	LPAS	TPAS

ma – my address has occurred
to – the talk-only mode is enabled
lo – the listen-only mode is enabled
ATN – the Attention command is asserted
TACS – GPIA is in the Talker Active State
LACS – GPIA is in the Listener Active State
LPAS – GPIA is in the Listener Primary Addressed State
TPAS – GPIA is in the Talker Primary Addressed State

ADDRESS SWITCH REGISTER R4R – The address switch register is external to the chip There is an enable line ($\overline{\text{ASE}}$) to be used to enable three-state drivers connected between the address switches and the MPU When the MPU addresses the address switch register the $\overline{\text{ASE}}$ line directs the switch information to be sent to the MPU. The five least-significant bits of the 8-bit register are used to specify the bus address of the device and the remaining three bits may be used at the discretion of the user The most probable use of one or two of the bits is for controlling the listener only or talk only functions

Address Switch Register
(Read Only)

UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1

AD1-AD5 – Device address
UD1-UD3 – User definable bits
When this "register" is addressed, the $\overline{\text{ASE}}$ pin is set which allows external address switch information from the bus device to be read

ADDRESS REGISTER R4W – The Address Register is an 8-bit storage register The purpose of this register is to carry the primary address of the device The primary address is placed in the five least-significant bits of the register. If external switches are used for device addressing these are normally read from the Address Switch Register and then placed in the Address Register by the MPU

AD1 through AD5 bits 0-5 are for the device's address The lsbe bit 7 is set to enable the Dual Primary Addressing Mode During this mode the device will respond to two consecutive addresses, one address with AD1 equal to 0 and the other address with AD1 equal to 1. For example, if the device's address is HEX 0F, the Dual Primary Addressing Mode would allow the device to be addressed at both HEX 0F and HEX 0E The dal bit 6 is set to disable the listener and the dat bit 5 is set to disable the talker

This register is cleared by the $\overline{\text{RESET}}$ input only (not by the reset bit of the Auxiliary Command Register bit 7).

When $\overline{\text{ATN}}$ is enabled and the primary address is received on the $\overline{\text{IB0-7}}$ lines, the MC68488 will set bit 7 of the address status register (ma). This places the MC68488 in the TPAS or LPAS.

When $\overline{\text{ATN}}$ is disabled the GPIA may go to one of three states. TACS, LACS, or SPAS

Address Register
(Write Only)

lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1

lsbe – enable dual primary addressing mode
dal – disable the listener
dat – disable the talker
AD1-AD5 – Primary device address, usually read from address switch register

Register is cleared by the $\overline{\text{RESET}}$ input pin only

AUXILIARY COMMAND REGISTER R3R/W – Bit 7, reset, initializes the chip to the following states (reset is set true by external $\overline{\text{RESET}}$ input pin and by writing into the register from the MPU).

SIDS – Source Idle State
AIDS – Acceptor Idle State
TIDS – Talker Idle State
LIDS – Listener Idle State
LOCS – Local State
NPRS – Negative Poll Response State
PPIS – Parallel Poll Idle State
PUCS – Parallel Poll Unaddressed to Configure State
PPO – Parallel Poll No capability

rfdr (release RFD handshake) bit 6 allows for completion of the handshake that was stopped by RFD (Ready For Data) holdoff commands hlda and hlde

fget (force group execute trigger) bit 0 has the same effect as the GET (Group Execute Trigger) command from the controller

rtl (return to local) bit 2 allows the device to respond to local controls and the associated device functions are operative

dacr (release DAC handshake) bit 4 is set high to allow DAC to go passively true This bit is set to indicate that the MPU has examined a secondary address or an undefined command

upla (upper/lower primary address) bit 1 will indicate the state of the LSB of the address received on the DIO1-8 bus lines at the time the last Primary Address was received This bit can be read but not written by the MPU

msa (valid secondary address) bit 3 is set true (high) when TPAS (Talker Primary Addressed State) or LPAS (Listener Primary Addressed State) is true The chip will become addressed to listen or talk. The primary address must have been previously received

RFD, $\overline{\text{DAV}}$, DAC — (Ready For Data, Data Valid, Data Accepted) bits assume the same state as the corresponding signal on the MC68488 package pins The MPU may only read this bit These signals are not synchronized with the MPU clock

dacd (data accept disable) bit 1 set high by the MPU will prevent completion of the automatic handshake on Addresses or Commands dacr is used to complete the handshake

feoi (forced end or identify) bit 5 tells the chip to send $\overline{\text{EOI}}$ low. The $\overline{\text{EOI}}$ line is then returned high after the next byte is transmitted NOTE· The following signals are not stored but revert to a false (low) level one clock cycle (MPUϕ2) after they are set true (high)

1 rfdr
2 feoi
3. dacr

These signals can be written but not read by the MPU

Auxiliary Command Register

reset	rfdr	feoi	dacr	msa	rtl	dacd	fget	Write
	DAC	DAV	RFE			ulpa		Read

reset — initialize the chip to the following status
(1) all interrupts cleared
(2) following bus states are in effect SIDS, AIDS, TIDS, LIDS, LOCS, PPIS, PUCS, and PPO
(3) bit is set by $\overline{\text{RESET}}$ input pin
msa — if GPIA is in LPAS or TDAS, setting msa will force GPIA to LADS or TADS
rtl — return to local if local lockout is disabled
ulpa — state of LSB of bus at last-primary-address receive time
fget — force group execute trigger command from the MPU has occurred
rfdr — continue handshake stopped by RFD holdoff
feoi — set EOI true, clears after next byte transmitted
dacr — MPU has examined an undefined command or secondary address
dacd — prevents completion of automatic handshake on Addresses or Commands

COMMAND STATUS REGISTER R1R — The command status register flags command or state as they occur These flags or states are simply coupled on the MPU bus There are five major address commands REM shows the remote/local state of the talker/listener REM bit 6, set low, implies the local state. LOK bit 5 shows the local lockout status of the talker/listener RLC bit 3 is set when a change of state of the remote/local flip-flop occurs and reset when the command status register is read DCAS bit 1 indicates that either the device clear or selected device clear has been received activating the device clear function SPAS bit 2 indicates that the SPE command has been received activating the device serial poll function UACG bit 7 indicates that an undefined address command has been received and depending on programming the MPU decides whether to execute or ignore it UUCG bit 0 indicates that an undefined universal command has been received

Command Status Register
(Read)

UACG	REM	LOK	X	RLC	SPAS	DCAS	UUCG

UACG — Undefined Addressed Command
REM — Remote Enabled
LOK — Local Lockout Enabled
RLC — Remote/Local State Changed
SPAS — Serial Poll Active State is in effect
DCAS — Device Clear Active State is in effect
UUCG — Undefined Universal Command

COMMAND PASS-THROUGH REGISTER R6R — The command pass through is an 8-bit port with no storage When this port is addressed by MPU it connects the instrument data bus ($\overline{\text{IB0}}$-$\overline{\text{IB7}}$) to the MPU data bus D0-D7 This port can be used to pass commands and secondary addresses that aren't automatically interpreted through to the MPU for inspection

Command Pass-Through Register
(Read Only)

B7	B6	B5	B4	B3	B2	B1	B0

An 8-bit input port used to pass commands and secondary addresses to MPU which are not automatically interpreted by the GPIA

PROGRAMMING CONSIDERATIONS

The following is a list of considerations when using the MH version of the MC68488

1 Handshake Interruption

Once a handshake sequence begins on the IEEE-488 bus it should be allowed to complete in a normal fashion, as described in the IEEE Standard If this sequence is interrupted (e g , the controller forces the DAV line to the not data valid state prematurely) the integrity of the data is lost and the interface devices can go to unintended states, as explained in the standard NOTE The MC68488 does not interrupt a handshake It always allows the handshake to complete in the correct sequence, however, it is possible for a device, other than a MC68488, connected to the bus to interrupt the sequence The controller can do this through an asynchronous Bus Take-over (asserting the ATN line during a handshake) If this occurs the controller should follow the asynchronous take-over with an IFC Uniline Command (ATN can be either asserted or not asserted at this time) It is also possible for some devices to interrupt the handshake by prematurely making the DAV line false (this type of interrupt should be avoided and it should be noted that the MC68488 does not interrupt the handshake sequence) If the DAV handshake line is made false (high) before DAC is made false (high) during a handshake sequence, the listener GPIA(s) will respond as follows

a) If IFC is sent by the controller with ATN false before another handshake sequence is initiated the MC68488 will reset back to an idle state The GPIA at this point is ready to be reprogrammed

The BI status bit may be set, depending on when the handshake was interrupted, but any byte in R7R cannot be considered valid. NOTE: If IFC is sent with ATN true, it must be sent again with ATN false

b) If another handshake is initiated before IFC is sent with ATN false, the GPIA does not generate interrupts for subsequent data bytes received by the listener GPIA(s). The device responds to commands and moves into and out of TACS, LACS, etc , but no further BI interrupts are generated The only solutions to this situation are to reset the MC68488 or have the MPU perform a read of R7R register in the GPIA

2 Interrupt Structure

The status bits in R0R, when set, cause an interrupt (drives the $\overline{IRQ}$ line low) if the appropriate interrupt mask bits in R0W are set. The $\overline{IRQ}$ line is sensitive to a low-to-high transition produced by the logical OR of the appropriate bits in R0R If, for example the BI status bit is set and causes an $\overline{IRQ}$ interrupt, the MPU reads R0R (this read will reset the $\overline{IRQ}$ line but not the status bit) and detect that the BI bit is set The software should then direct the MPU to read the data byte from R7R, which in turn causes the BI bit to be reset If after the status register (R0R) was read and before R7R is read, another interrupt status bit is set (e.g , the CMD bit) this second condition does not cause an interrupt The BI bit being set at the time CMD occurred prevents the $\overline{IRQ}$ line from detecting the necessary low-to-high transition and an interrupt could be missed. To prevent this, the last set of instructions in the software interrupt handler should be a reset of the interrupt mask register, followed by programming this same register to its original state This always produces the needed low-to-high transition, preventing missed interrupts

3 The "nba" for TACS affects "nba" for SPAS

If nba for TACS is false (there is not a data byte pending in R7W) then the serial poll handshake sequence for the status byte to the controller occurs once If nba for TACS is true (there is a data byte in R7R waiting for a handshake to listeners) then the status byte will be sent to the controller each time the controller completes a handshake and indicates that it is ready for more data

4 The "nba" for SPAS affects "nba" for TACS

The controller places the GPIA into the Serial Poll Active State (SPAS) by sending serial Poll enable, sending the device talk address, and then releasing ATN If the controller does this and never accepts the serial Poll Status byte (never makes the RFD handshake line true) but rather the controller asserts ATN and sends Serial Poll Disable (SPD), then the GPIA moves into and out of SPAS without completing the status byte handshake routine In this state the "nba" for SPAS remains true and affects "nba" for TACS in the following way

When the controller places the GPIA in TACS the part makes DAV true as soon as RFD is made true by the listeners in a normal sequence However, the GPIA continues the handshake sequence, using the contents of R7W, over and over. (i.e., Each time the listeners accept the current data byte and makes RFD true, the GPIA makes DAV true automatically and begins another handshake sequence) The B0 status bit is set, however, and if the MPU writes to R7W the new data byte is sent to the listeners over and over, using a handshake routine This continual sending of data bytes occurs until the controller places the GPIA back in SPAS and completes the handshake routine for the Serial Poll Status byte making "nba" for SPAS false again

This situation does not occur if the controller handshakes the status byte when it places the GPIA in SPAS

5 Dual Addressing

Dual addressing implies the use of two adjacent primary addresses and, as such, care should be taken when selecting the primary addresses for this mode Decimal address 30 (11110) should not be used because the dual address counterpart of decimal 30 is decimal 31 (11111) Since address 31 has the same bit code as that of either the Untalk or Unlisten Commands this value is an invalid primary address for the IEEE-488 system

6 "Ghost Interrupts"

A "ghost interrupt" is an interrupt that occurs as a result of the MC68488, but when the status register is checked no status bits are set There are two conditions that can legitimately cause a "ghost interrupt " They are

a) SPAS status bit

If the controller conducts a serial poll by sending Serial Poll Enable (SPE) and then sends the GPIA talk address, the SPAS status bit is set and can cause an interrupt After the controller receives the Serial Poll Status byte it will send Serial Poll Disable (SPD) which resets the SPAS status bit If the controller can perform this sequence of events before the interrupt handler can check the SPAS status bit, the MPU will not find any status bits set ("ghost interrupt"). The possibilities are twofold

1) If this device had actually requested the service, then the MPU, after receiving the interrupt ("ghost" or not), should check bit 6 of the Serial Poll register If this bit is reset the MPU knows that a Serial Poll was conducted and can reset the rsv as per normal Serial Poll handling procedures

2) If this device did not request the service request and SPAS is not set, the software should detect this as a "ghost interrupt," ignore it, and proceed with normal operations

See "Serial Poll Procedure" (#11) for further Serial Poll operation

b) B0 status bit

The B0 Status bit is set whenever the MC68488 is in the Talker Active State and the output register (R7W) is empty. After the listener(s) accept the current data byte on the IEEE bus, the B0 status bit will again be set and with the appropriate mask bits set, will cause an interrupt. When the talker sends the last byte of a string it is possible for the controller to detect this, synchronously take control of the bus, and untalk the talker, however, when the last byte is accepted the B0 status bit is again set and if so programmed, causes another interrupt. It is possible for the controller to untalk the device thereby resetting B0 before the MPU interrupt handler is able to check the status register. Under these conditions a "ghost interrupt" occurs. See "Send Last Byte Procedure" (#10) for further description and solution.

7. UACG Status Bit

The UACG status bit is set anytime the GPIA receives an Undefined Address Command Group (UACG) message from the controller. This bit is not qualified with the addressed state of the part. The MPU software, after detecting a UACG, must check the ma bit in the address status register to see if the device is addressed. If the UACG message is a selected command only pertinent to addressed listeners, the software, after receiving the command by reading R6R, should release the handshake (write dacr high in R3W). This allows the controller to make ATN false. If the device has been addressed to listen/talk and ATN is made false the LACS/TACS status bit in the address status register will be set. The MPU can then check these bits.

8. END Status Bit

The END status bit in R0R is used to indicate to addressed listeners that the next byte received by the addressed talker is the last byte of a string. This bit is not qualified with the handshake and thus occurs ahead of the reception of the last data byte. This alerts the MPU that the final byte will soon follow. Because of this, two interrupts, if so programmed, will occur. One for the END bit and one for the BI bit when the final byte is transferred with a handshake. For those situations where it is inconvenient to have two interrupts the END status bit can be masked, not allowing it to cause an interrupt.

9. feoi (force end or identify)

This control bit (bit 5, R3W) is used when the MC68488 is an Active Talker, to indicate to the listener(s) on the IEEE bus the end of a data string transfer. The MC68488 asserts the EOI management line when the feoi control bit is set and the device is in the Talker Active State (TACS). The feoi bit is set by the MPU writing this bit high and automatically resets one E clock cycle after it was set. The use of this function is as follows. When sending a string of data the feoi control bit should be set prior to sending the final data byte. This causes the EOI management line to be asserted (low). The final data byte can now be sent. The EOI line remains asserted until this byte is accepted, at which time it returns high.

Care must be used when setting the feoi control bit. Once feoi has been written high, the EOI line is asserted when the MC68488 is an Active Talker and remains asserted until the next data byte is sent and accepted. This is true even if feoi is written high while the device is not an Active Talker. In this case the EOI management line is asserted as soon as the MC68488 is again made an Active Talker. Once the feoi control bit is set, only a device reset prevents the END message from being sent when the MC68488 becomes an Active Talker.

10. Send Last Byte Procedure (Talker Mode)

The procedure used for sending the last byte is described below. When using the $\overline{\text{EOI}}$ management line, the MPU software must first set the feoi control bit (asserting $\overline{\text{EOI}}$), and then send the last byte. When the last byte is accepted by all listeners, the B0 status bit of the talker device is set. The B0 status bit is not qualified with the $\overline{\text{EOI}}$ line, but is set whenever the current data byte is accepted by all listeners and the device is in the Talker Active State (TACS). (Note that when the controller asserts $\overline{\text{ATN}}$ to send commands, the GPIA moves out of TACS causing B0 to reset and remains out of TACS as long as $\overline{\text{ATN}}$ is asserted.) After the data block transfer, the controller takes control of the bus (asserts $\overline{\text{ATN}}$) and reconfigures the GPIB system. In performing this task, the controller sends command(s) that untalk the device (MLA, OTA, UNT) or reassigns it as a Talker (MTA) asking for further data transfers. Since the GPIB operates asynchronously with respect to the device MPU bus, it is possible for the controller to take control of the GPIB and cause actions that change the state of the B0 status bit in the middle of the MPU interrupt routine. As a result, care needs to be exercised when responding to the B0 status bit interrupt occurring after transferring the last byte. Any of the following conditions can occur.

1) Device Untalked – If either My Listen Address, Other Talk Address, or the Untalk command is sent, the device is placed in the Talker Idle State (TIDS) – the device is Untalked. In this case the B0 status bit is set as soon as the last data byte is accepted, reset when the controller asserts $\overline{\text{ATN}}$, and B0 will remain reset after $\overline{\text{ATN}}$ is released.
   (a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{\text{ATN}}$. This status indication, however, is misleading as another byte transfer is not intended. The device is soon to be Untalked.
   (b) The B0 status bit indicates a reset condition if the MPU reads the Interrupt Status Register after $\overline{\text{ATN}}$ has been asserted – a "ghost interrupt" is produced. This B0 status bit remains reset after $\overline{\text{ATN}}$ is made false (high).
2) Device Reassigned as a Talker – The controller reassigns the device to talk by sending My Talk Address. In this case the B0 status bit is set as

soon as the last data byte is accepted, reset when the controller asserts $\overline{\text{ATN}}$ to send MTA, and is again set when $\overline{\text{ATN}}$ is made false by the controller

- (a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{\text{ATN}}$. This case is identical to part (a) for "Device Untalked" shown above.
- (b) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register while $\overline{\text{ATN}}$ is asserted – a "ghost interrupt" is produced
- (c) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register after $\overline{\text{ATN}}$ is made false (high) This status indication is requesting a byte transfer and should be acted upon accordingly

To alleviate the above ambiguity and "ghost interrupt" situation, the GPIB handshake must be synchronized with action by the device MPU The following step-by-step procedure provides this needed synchronization and eliminates the ambiguity when servicing the B0 status bit after sending the last byte

1) Before sending the last byte of a block transfer, the feoi bit (if used) should be set. In addition, the dacd bit in R3W should be set, holding off the handshake upon reception of any command (establishes the required synchronization between MPU and controller).
2) If operating under interrupts, the B0 interrupt mask should be reset. This prevents generation of a B0 status interrupt when the last byte is received.
3) Send the last data byte
4) The MPU now monitors the ATN bit in the Address Status Register (R2R) When the ATN bit is set, the $\overline{\text{ATN}}$ line has been asserted and it will remain asserted until completion of the handshake The procedure, described herein, assumes that $\overline{\text{ATN}}$ line is asserted between block transfers and at least one command sent The fact that $\overline{\text{ATN}}$ is asserted indicates that the device is no longer in TACS and, thus, the B0 status bit is reset
5) The dacd bit in R3W can now be written low, removing the manual handshake hold-off on subsequent commands With the same write instruction, the dacr bit should be set, releasing the handshake on the current command (write a hex 10 to R3W)
6) The B0 interrupt mask bit can now be set, enabling interrupts for another block transfer.

After following this procedure, a B0 status condition will occur only if a second block of data is requested by the controller In addition, the possibility of a B0 "ghost interrupt" is eliminated

11 Serial Poll Procedure

The MPU initiates a service request by writing rsv (bit 6, R5W) high in the GPIA At the same time, the appropriate code should be placed in the other 7 bits Bit 6 being set causes the $\overline{\text{SRQ}}$ management line to go low. The GPIA enters the Serial Poll Active State (SPAS) when it receives SPE and is an active talker When it enters SPAS, the following occurs: the SPAS status bit (bit 2, R1R) is set, the CMD status bit (bit 2, R0R) is set, the $\overline{\text{SRQ}}$ line is asserted passively false (high), the SRQ status bit (bit 6, R5R) is reset, and the contents of R5R is placed on the GPIB data bus

When the GPIA enters SPAS, the SPAS status bit (R1R) is set. This, in turn, causes the CMD status bit in R0R to be set. In an interrupt driven system with the CMD and IRQ mask bits set, this causes an MPU interrupt. These status bits are not latched conditions and only monitor the current state of the GPIA If the controller places the GPIA in SPAS (sends SPE and MTA), receives the Serial Poll status byte and removes the GPIA from SPAS (sends SPD) before the MPU reads the Interrupt Status register, the contents of this register shows hex 10 Since the MPU knows that this device issued the service request, it should check bit 6 of R5W if an MPU interrupt is generated but no status bit is set If bit 6, R5R, is reset, the MPU will know the controller has performed a Serial Poll on it. However, the SRQ status bit being reset does not indicate that the status byte was accepted by the controller – that is, the handshake was completed. Rather, it indicates that the GPIA has been placed in SPAS and that the status byte has been placed on the GPIB In systems with slow responding controllers, the SRQ bit in R5R can be reset while the SPAS status bit is still set In this case to determine when the status byte was accepted, the MPU can monitor SPAS status bit This bit is reset when the controller has removed the GPIA from the SPAS. Once in SPAS, the controller must accept the Serial Poll byte before removing the device from SPAS The rsv bit cannot be written low until the status byte has been accepted, but should be written low as soon as the status byte has been accepted by the controller.

If this device has issued a service request to the controller, the following provides a procedure for handling a SPAS interrupt The procedure only discusses Serial Poll (SPAS) interrupts Interrupts resulting from other sources need to be incorporated as appropriate for the system application In an interrupt driven system, the MPU normally reads the Interrupt Status Register to find the cause of the interrupt The Interrupt Status Register must be read to release the $\overline{\text{IRQ}}$ line and, in most cases, it will be read to check if something other than SPAS caused the interrupt However, since it is possible that the SPAS status can be set and then reset before the MPU reads the register, the following procedure should also be used (even though the SPAS status is reset)

1) The MPU should monitor the SRQ bit in the Serial Poll Register This can occur as a result of either an interrupt or a polling routine
2) When the SRQ bit returns to zero, it indicates that the MC68488 has been placed in the Serial Poll Active State (SPAS). This does not mean that the device is in SPAS, because the con-

troller could have placed the MC68488 in SPAS and then removed the device from SPAS before the MPU reads the Serial Poll Register (R5R)

3) After the SRQ bit in R5R returns to zero, the MPU should read the Command Status Register and Monitor the SPAS status bit. When this bit returns to 0, it indicates that the Serial Poll Status byte has been accepted by the controller and that the MC68488 has been removed from the Serial Poll Active State (SPAS)
4) After the SPAS status bit returns to 0, the rsv bit (in R5W) should be written low

The GPIA uses the source handshake to send the Serial Poll status byte to the controller The GPIA does this by placing the status byte on the GPIB, and when the controller makes RFD true, the GPIA makes $\overline{\text{DAV}}$ true (low), and the handshake takes place according to the IEEE-488 Standard handshake protocol If nba for the GPIA TACS function is false at this time, the GPIA will send this byte only once, i.e., the GPIA does not make $\overline{\text{DAV}}$ true (low) a second time If nba for the GPIA TACS function is true at this time, the GPIA sends this byte over and over, provided the controller continually makes RFD true at the end of the handshake without reconfiguring the device, i.e , the GPIA in this situation makes $\overline{\text{DAV}}$ true (low) each time it receives an RFD true from the controller The only time nba can be true for TACS is if the device was an active talker prior to the Serial Poll sequence, and the GPIA MPU had loaded a byte in R7W Now, if the controller synchronously takes over the bus before this byte is placed on the GPIB, the nba for TACS will be true

NOTE

After a Serial Poll has been conducted on the GPIA and the SRQ bit (bit 6, R5W = 0) is reset, the MPU must write the rsv (bit 6, R5W) low before another service request can be initiated.

APPENDIX

GPIA MASK SET DIFFERENCES

There have been two mask sets produced for the GPIA (MC68488). They are·

G6G MASK SET — sampled in the fall of '77 (first mask set). This mask set was produced through December of 1978 and can be identified by the letters "GG" preceding the date code on top of the package.

M2H MASK SET — parts available January '79 (final mask set). Any parts ordered after this date will be M2H parts The M2H mask set replaces the G6G mask set and can be identified by the letters MH or M2H preceding the date code on top of the package The mask set designation for later production runs is P9W. The P9W mask set is identical to the M2H in all aspects.

There are seven areas of differences between the G6G and M2H/P9W mask sets. They are·

1 RLC Status bit

This bit is used to implement the Remote/Local interface function. In the GG mask set the Remote/Local option should not be used, because the RLC status bit in R1R will lock up in the zero state In the MH mask version the RLC bit is completely functional and will report any change in the REM status bit

2 Extended Addressing

The GG mask version of the GPIA will not discontinue secondary addressing when the primary address of another GPIA is sent by the controller, i e , after entering LPAS, the primary address of another device will not transfer the GPIA to LPIS This transition from LPAS to LPIS was not fully implemented in the GG mask set The MH mask set has fully implemented this interface function With this version, if the GPIA is programmed for extended addressing and receives its primary address, it will move to LPAS If at this point the primary address of another controller is sent, the GPIA will go to its idle state (LIDS/TIDS) as per IEEE-488 1978 standard requirements

3. TPAS and LPAS Status Bits

In the GG mask set the LPAS status bit will report either LPAS or LADS Likewise, the TPAS status bit will report either TPAS or TADS In the MH mask set these bits only report LPAS and TPAS respectively

4 DAC Release

When the GPIA (GG mask set) in the listener mode receives a byte of data from the IEEE bus the DAC handshake will be held off until the MPU reads this data byte The E-pulse that reads this data from R7R also releases DAC, indicating to the talker that the byte has been accepted In the GG mask set the DAC handshake line is released on the low-to-high transition (leading edge) of the E-pulse, but the data is actually read (accepted by the MPU) on the high-to-low transition (trailing edge) If there is a very fast talker or long E-pulse width, it is possible for the talker to receive a data accept (DAC) and place the next data byte in the Data-In Register before the current one has been read out by the MPU This will overwrite the data on the MPU bus and result in missed data For this to occur the talker must be able to detect the DAC line going high and place the next data byte on the bus (making DAV true) before the E-pulse goes low For a 1 MHz E-pulse this is approximately 400 ns The MH or M2H mask set corrects this by releasing DAC after the trailing edge of the E-pulse

5 dsel (deselect)

One of the functions of the dsel bit (bit 7 of R2W) is to deselect the Group Execute Trigger (GET) command from setting the GET status bit and causing an interrupt. The GG mask version of the GPIA, when dsel is set, prevents the GET status bit from being set, but it is still possible to get an $\overline{\text{IRQ}}$ ($\overline{\text{IRQ}}$ output goes low), if enabled, when the GET command is detected Thus, dsel inhibits the GET status condition, but not the associated interrupt The result is a "ghost interrupt" whenever the controller sends the GET command. The MH mask set, when in dsel mode, inhibits both GET status and its associated interrupt and eliminates this "ghost interrupt."

6 hold-on-all-data (hlda)

When in the listener mode, the GPIA provides a means of holding off the handshake on reception of data until the MPU releases the handshake This mode occurs if the hlda (hold-on-all-data) bit in R2W is set The MPU releases the handshake by writing rfdr (ready-for-data-release) in R3W high In the GG mask version, if while receiving data in the listener mode the controller takes over synchronously and makes the GPIA a talker and then changes the GPIA at a later time, back to a listener, the RFD handshake will be held off on the listener command rather than waiting for the first data byte The MH mask only holds off the handshake on data and does not hold off the handshake on any command

7 new-byte-available (nba) During a Serial Poll

In the GG mask version, if the GPIA had been in the talker active state prior to the controller conducting a serial poll, it is possible for nba to be lost The situation is as follows: if a byte of data has been written into R7W and the controller takes over the bus synchronously at SDYS (SH state diagram, Figure 3, page 20, IEEE-488 1978 Specification) to perform a serial poll, the GG mask version of the GPIA will respond in one of two ways

a) If the controller never requests the contents of the Serial Poll Register from the GPIA, and when the GPIA is returned as a talker, the data in R7W is available to the listeners on the bus (Data byte is not destroyed)

b) If the controller requests the contents of the Serial Poll Register from the active talker, when the controller returns the GPIA to the Active Talker State, the data which was in R7W will have been handshaked as though it had been accepted by the active listeners (data byte will be destroyed)

The original IEEE Standard had a discrepancy as to what happens to this data byte (nba) under these circumstances This discrepancy has been alleviated The MH mask conforms to the latest revision and does not destroy the data in R7W when a Serial Poll occurs, i e , if in TACS with a nba pending when the controller releases the bus to the talker, the byte in R7W will be transferred, via handshake, to the listeners (data byte is not destroyed)

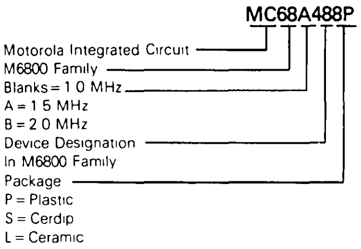
SOFTWARE DIFFERENCES BETWEEN MASK SETS

The seven changes mentioned in the previous sections are the only changes from the GG to the MH mask set All of these changes except number 3 (TPAS and LPAS status bits) are transparent to the user software.

The change to TPAS and LPAS status bits is a functional change. In the GG mask, user software could monitor LPAS and TPAS for address recognition in the primary address mode because LPAS is set as soon as the GPIA receives its Listen Address (MLA) and TPAS is set as soon as the GPIA receives its Talker Address (MTA), i e , the LPAS bit is set when the GPIA enters LADS and the TPAS bit is set when the GPIA enters TADS. In the MH mask set these bits do not report LADS/TADS and as such they can only be used in the extended address mode In the primary address mode the software for the MH mask set should monitor LACS/TACS (bits 2 and 3 of the address status register) rather than LPAS/TPAS. TACS/LACS indicates when the device is in the Talker/Listener Active State

MECHANICAL DATA

ORDERING INFORMATION



BETTER PROGRAM

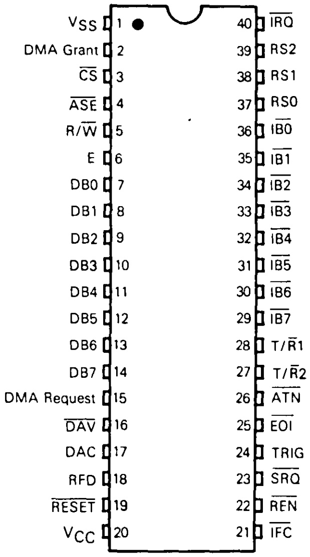
Better program processing is available on all types listed Add suffix letters to part number

Level 1 add "S" Level 2 add "D" Level 3 add "DS"

Speed	Device	Temperature Range
1.0 MHz	MC68488P, S MC68488CP, CS	0°C to 70°C −40°C to +85°C
1.5 MHz	MC68A488P, S MC68A488CP, CS	0°C to 70°C −40°C to +85°C
2.0 MHz	MC68488P, S	0 to +70°C

Level 1 "S" = 10 Temp Cycles – (−25 to 150°C),
Hi Temp testing at T_A max
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2

PIN ASSIGNMENT


V_{SS} 1
DMA Grant 2
$\overline{CS}$ 3
$\overline{ASE}$ 4
R/$\overline{W}$ 5
E 6
DB0 7
DB1 8
DB2 9
DB3 10
DB4 11
DB5 12
DB6 13
DB7 14
DMA Request 15
$\overline{DAV}$ 16
DAC 17
RFD 18
$\overline{RESET}$ 19
V_{CC} 20
40 $\overline{IRQ}$
39 RS2
38 RS1
37 RS0
36 $\overline{IB0}$
35 $\overline{IB1}$
34 $\overline{IB2}$
33 $\overline{IB3}$
32 $\overline{IB4}$
31 $\overline{IB5}$
30 $\overline{IB6}$
29 $\overline{IB7}$
28 T/$\overline{R}$1
27 T/$\overline{R}$2
26 $\overline{ATN}$
25 $\overline{EOI}$
24 TRIG
23 $\overline{SRQ}$
22 $\overline{REN}$
21 $\overline{IFC}$

MC6898

Technical Summary
Cable Driver/Receiver

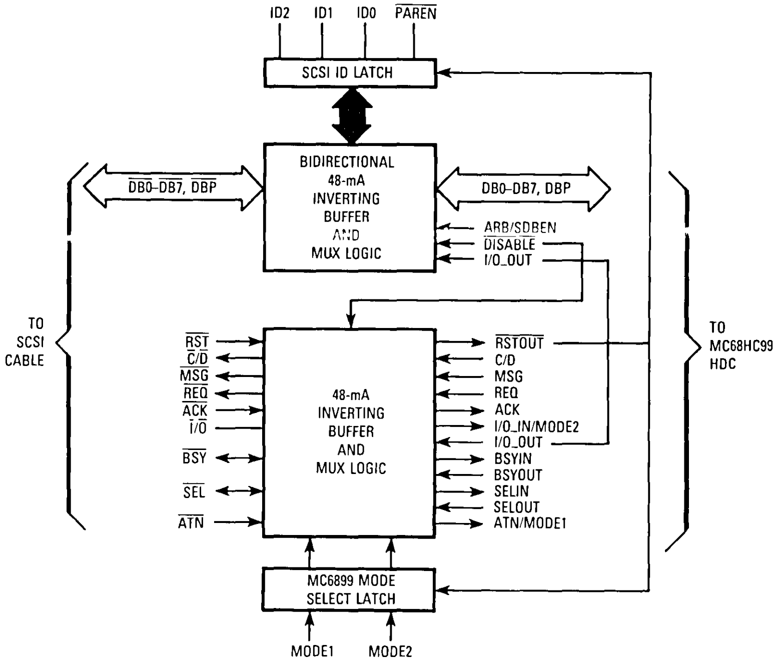
The MC6898 is a peripheral device providing single-ended, 48-mA, small computer system interface (SCSI) cable drivers/receivers in a single chip. The MC6898 is used to interface the MC68HC99 hard disk controller (HDC) to an SCSI bus, enhancing its effectiveness as an HDC (see Figure 1). In addition, mode, parity, and encoded ID inputs to the MC6898 are used to initialize the MC68HC99 after reset. The ID inputs are also decoded and driven onto the SCSI bus during arbitration.

The MC6898 conforms to all requirements of the ANSI X3T9.2 SCSI standard. Use of the MC6898 greatly reduces the number of chips needed to interface the MC68HC99 to other SCSI devices.

This technical summary contains limited information on the MC6898. Major hardware features of the MC6898 SCSI receiver/driver are as follows:

- 48-mA Single-Ended Drivers/Receivers
- 52-Pin PLCC Package for Shock and Vibration Resistance
- Internal Logic to Drive the MC6899 Mode Select Signals, SCSI ID Signals, and Parity Disable Signal at Reset

BLOCK DIAGRAM


ID2
ID1
ID0
$\overline{\text{PAREN}}$
SCSI ID LATCH
BIDIRECTIONAL 48-mA INVERTING BUFFER AND MUX LOGIC
$\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$
DB0–DB7, DBP
ARB/$\overline{\text{SDBEN}}$
$\overline{\text{DISABLE}}$
I/O_OUT
TO SCSI CABLE
TO MC68HC99 HDC
$\overline{\text{RST}}$
$\overline{\text{C/D}}$
$\overline{\text{MSG}}$
$\overline{\text{REQ}}$
$\overline{\text{ACK}}$
$\overline{\text{I/O}}$
$\overline{\text{BSY}}$
$\overline{\text{SEL}}$
$\overline{\text{ATN}}$
48-mA INVERTING BUFFER AND MUX LOGIC
$\overline{\text{RSTOUT}}$
C/D
MSG
REQ
ACK
I/O_IN/MODE2
I/O_OUT
BSYIN
BSYOUT
SELIN
SELOUT
ATN/MODE1
MC6899 MODE SELECT LATCH
MODE1
MODE2

This document contains information on a new product. Specifications and information herein are subject to change without notice.

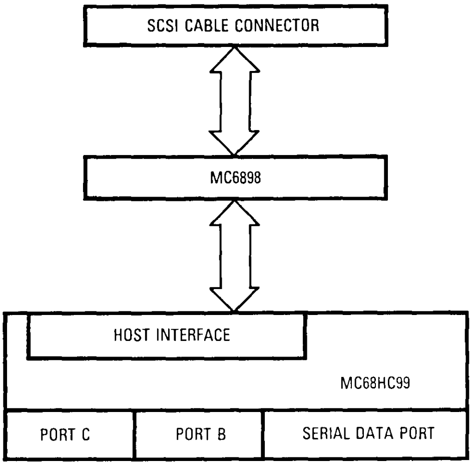


Figure 1. Basic SCSI Interface

SIGNAL DESCRIPTION

The following paragraphs contain a description of the MC6898 signals.

V_{DD}

V_{DD} supplies +5.0 Vdc power to the MC6898. The actual operational range of the power supply is specified in the maximum ratings table (see **ELECTRICAL CHARACTERISTICS**).

V_{SS}

V_{SS} pins provide a current return path for the MD6898 power supply. All signal levels are referenced to V_{SS}.

RESET ($\overline{RST}$, $\overline{RSTOUT}$)

$\overline{RST}$ is an input signal that, when asserted, causes DB0–DB7, $\overline{DBP}$ and DB3–DB7 to be driven to their quiescent state. The assertion of $\overline{RST}$ also forces the values on the ID0–ID2, $\overline{PAREN}$, MODE1, and MODE2 pins to be driven on the DB0–DB2, $\overline{DBP}$, ATN/MODE1, and I/O_IN/MODE2 pins, respectively. $\overline{RSTOUT}$ is an output signal that reflects the value of the $\overline{RST}$ pin.

SCSI BUS ID (ID0–ID2)

ID0–ID2 are input signals used to determine the SCSI device ID of the MC68HC99. When $\overline{RST}$ is asserted, the values on the ID0–ID2 pins are latched by the MC6898 and driven onto the DB0–DB2 pins, respectively. The SCSI ID is determined according to Table 1.

Table 1. HDC SCSI Bus ID

ID2	ID1	ID0	Bus ID
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

NOTE: In columns 1–3, a value of zero equals logic low; one equals logic high.

MODE SELECT (MODE1, MODE2)

MODE1 and MODE2 are input signals used to determine the operating mode of the MC68HC99. When $\overline{RST}$ is asserted, the values on the MODE1 and MODE2 pins are driven onto the ATN/MODE1 and I/O_IN/MODE2 pins, respectively. The operating mode of the MC68HC99 is then determined according to Table 2.

Table 2. HDC Mode Selection

MODE1 Pin	MODE2 Pin	Mode Selection
0	0	Expanded 3
0	1	Expanded 2
1	0	Expanded 1
1	1	Single-Chip

NOTE: In columns 1 and 2, a value of zero equals logic low; one equals logic high.

PARITY SELECT ($\overline{\text{PAREN}}$)

$\overline{\text{PAREN}}$ is an input signal used to enable/disable the parity check function of the MC68HC99. When $\overline{\text{RST}}$ is asserted, the value on the $\overline{\text{PAREN}}$ pin is driven on the DBP pin. Parity is enabled when $\overline{\text{PAREN}}$ is logic low. Data bus parity is always odd.

SCSI DISABLE ($\overline{\text{DISABLE}}$)

$\overline{\text{DISABLE}}$ is an input signal used to drive the $\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$, and $\overline{\text{REQ}}$ pins to their inactive states. When $\overline{\text{DISABLE}}$ is asserted, the values on the C/D, MSG, and I/O_OUT pins are latched and driven (inverted) on the $\overline{\text{C/D}}$, $\overline{\text{MSG}}$, and $\overline{\text{I/O}}$ pins, respectively. Also, the level on the ACK pin reflects the level on the REQ pin when $\overline{\text{DISABLE}}$ is asserted.

SCSI DATA BUS ($\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$, DB0–DB7, DBP)

$\overline{\text{DB0}}$–$\overline{\text{DB7}}$ and $\overline{\text{DBP}}$ are open-collector-type, bidirectional signals. DB0–DB7 and DBP are also bidirectional signals. These signals collectively form the SCSI data bus; their function is determined in Table 3.

SELECT ($\overline{\text{SEL}}$, SELIN, SELOUT)

$\overline{\text{SEL}}$ is an open-collector-type, bidirectional signal. SELIN and SELOUT are output and input signals, respectively. These signals collectively form the SCSI SELECT signal. The SELIN pin always reflects the inverted level of the $\overline{\text{SEL}}$ pin. The level on the SELOUT pin is inverted and driven onto the $\overline{\text{SEL}}$ pin. The $\overline{\text{SEL}}$ pin, as an open-collector output, can only drive to the logic-low level. A terminating resistor should be used on $\overline{\text{SEL}}$. Refer to the ANSI X3T9.2 SCSI specifications for proper use of terminating resistors.

BUSY ($\overline{\text{BSY}}$, BSYIN, BSYOUT)

The relationship among the BUSY signals, $\overline{\text{BSY}}$, BSYIN, and BSYOUT, is exactly the same as the relationship described in **SELECT ($\overline{\text{SEL}}$, SELIN, SELOUT)**.

INPUT/OUTPUT ($\overline{\text{I/O}}$, I/O_IN/MODE2, I/O_OUT)

The relationship among the I/O signals, $\overline{\text{I/O}}$, I/O_IN/MODE2, and I/O_OUT is exactly the same as the relationship described in **SELECT ($\overline{\text{SEL}}$, SELIN, SELOUT)**. The single exception is that when $\overline{\text{RST}}$ is asserted, I/O_IN/MODE2 reflects the value of the MODE2 pin.

COMMAND/DATA (C/D, $\overline{\text{C/D}}$)

$\overline{\text{C/D}}$ is an open-collector-type output signal used on the SCSI cable. C/D is an input signal. The level on the C/D pin is inverted and driven on the $\overline{\text{C/D}}$ pin. Since it is an open-collector-type signal, $\overline{\text{C/D}}$ can only drive logic low. A terminating resistor should be used on $\overline{\text{C/D}}$. Refer to the ANSI X3T9.2 SCSI specifications for proper use of terminating resistors.

MESSAGE (MSG, $\overline{\text{MSG}}$)

The relationship between the MESSAGE signals, MSG and $\overline{\text{MSG}}$, is exactly the same as that described in **COMMAND/DATA (C/D, $\overline{\text{C/D}}$)**.

REQUEST (REQ, $\overline{\text{REQ}}$)

The relationship between the REQUEST signals, REQ and $\overline{\text{REQ}}$, is exactly the same as that described in **COMMAND/DATA (C/D, $\overline{\text{C/D}}$)**.

ACKNOWLEDGE (ACK, $\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ is an open-collector-type input signal used on the SCSI cable. ACK is an output signal that reflects the inverted value on the $\overline{\text{ACK}}$ pin. When $\overline{\text{DISABLE}}$ is asserted, ACK reflects the state of the REQ pin. A terminating resistor should be used on $\overline{\text{ACK}}$. Refer to the ANSI X3T9.2 SCSI specifications for proper use of terminating resistors.

ARBITRATION/SCSI DATA BUS ENABLE (ARB/SDBEN)

ARB/SDBEN is an input signal which, in conjunction with $\overline{\text{RST}}$, $\overline{\text{DISABLE}}$, and I/O_OUT, defines the function of the SCSI data bus ($\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$ and DB0–DB7, DBP). This function is clearly shown in Table 3.

ATTENTION ($\overline{\text{ATN}}$, ATN/MODE1)

$\overline{\text{ATN}}$ is an open-collector-type signal used on the SCSI cable. ATN/MODE1 is an output signal that reflects the inverted value on the $\overline{\text{ATN}}$ pin. When $\overline{\text{RST}}$ is asserted, ATN/MODE1 reflects the value of the MODE1 pin.

Table 3. SCSI Data Bus Signal Functions

Determining Signal States				Functions of $\overline{\text{DB0–DB7}}$, $\overline{\text{DBP}}$, DB0–DB7 and DBP[2]
$\overline{\text{RST}}$	$\overline{\text{DISABLE}}$	I/O_OUT	$\overline{\text{ARB}}$[1]	
0	X	X	X	Inactive: $\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$, and DB3–DB7. DB0–DB2 reflect values of ID0–ID2. DBP reflects the value of $\overline{\text{PAREN}}$.
1	0	X	X	Inactive: $\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$, DB0–DB7, and DBP
1	1	0	0	Values of $\overline{\text{DB0}}$–$\overline{\text{DB7}}$ and $\overline{\text{DBP}}$ are inverted and driven onto DB0–DB7 and DBP, respectively.
1	1	0	1	One pin of $\overline{\text{DB0}}$–$\overline{\text{DB7}}$ is asserted.[3] Inactive: All others.
1	1	1	0	Values of DB0–DB7 and DBP are inverted and driven onto $\overline{\text{DB0–DB7}}$ and $\overline{\text{DBP}}$, respectively.
1	1	1	1	Illegal Combination

NOTES:

1. In Table 3, zero equals logic low; one equals logic high; and X indicates that the value does not matter.
2. $\overline{\text{DB0–DB7}}$ and $\overline{\text{DBP}}$ are open-collector-type pins. Only a logic-low level can be driven. Terminating resistors should be used on these signals. Refer to the ANSI X3T9.2 SCSI specifications for proper use of terminating resistors.
3. The decoded value of ID0–ID2 determines which pin of $\overline{\text{DB0–DB7}}$ is asserted. If the combined hexadecimal value of ID0–ID2 is zero, then $\overline{\text{DB0}}$ is asserted. If the decoded value is two, then $\overline{\text{DB2}}$ is asserted, etc.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to V_{DD} +0.5	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance PLCC	θ_{JA}	50	°C/W

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 5%; V_{SS} = 0 Vdc; T_A = 0 to 70°C)

Characteristic	Symbol	Min	Max	Unit
Output Voltage $\overline{DB0}$–$\overline{DB7}$, $\overline{DBP}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$, $\overline{I/O}$, $\overline{BSY}$, $\overline{SEL}$ I_{Load} = 48 mA	V_{OL}	—	0.4	V
DB0–DB7, DBP, $\overline{RSTOUT}$, ACK, I/O_IN/MODE2, BSYIN, SELIN, ATN/MODE1 I_{Load} = 1.6 mA	V_{OL}	—	0.4	
I_{Load} = −800 μA	V_{OH}	2.4	—	
Input High Voltage	V_{IH}	2.0	—	V
Input Low Voltage	V_{IL}	—	0.8	V
Three-State Leakage $\overline{DB0}$–$\overline{DB7}$, $\overline{DBP}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$, $\overline{I/O}$, $\overline{BSY}$, $\overline{SEL}$	I_{OZ}	20	200	μA
Input Leakage $\overline{RST}$, $\overline{ACK}$, $\overline{ATN}$, ID0–ID2, C/D, $\overline{PAREN}$, ARB/SDBEN, $\overline{DISABLE}$, I/O_OUT, MSG, REQ, BSYOUT, SELOUT, MODE1, MODE2	I_{in}	—	±10	μA
Input Capacitance	C_{in}	—	15	pF
Power Dissipation (excluding I/O port)	P_D	—	50	mW

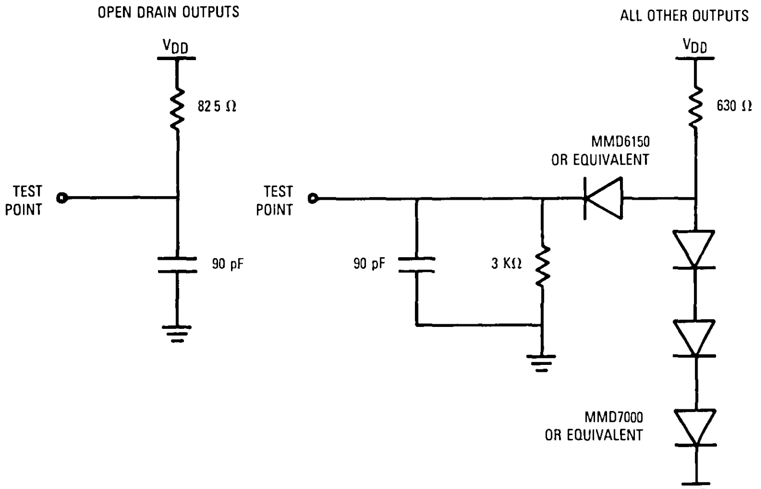


Figure 2. Test Load

TRANSITION TIMING (see Figure 3)

Timing Power Dissipation (t_{PD})			
Input	**Output**	**Max**	**Unit**
SELOUT	$\overline{\text{SEL}}$	30	ns
BSYOUT	$\overline{\text{BSY}}$	30	ns
I/O_OUT	$\overline{\text{I}}/\overline{\text{O}}$	30	ns
MSG	$\overline{\text{MSG}}$	30	ns
C/D	$\overline{\text{C}}/\overline{\text{D}}$	30	ns
$\overline{\text{SEL}}$	SELIN	40	ns
$\overline{\text{BSY}}$	BSYIN	40	ns
REQ	$\overline{\text{REQ}}$	40	ns
$\overline{\text{ACK}}$	ACK	50	ns
$\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$	DB0–DB7, DBP	80	ns
DB0–$\overline{\text{DB7}}$, DBP	$\overline{\text{DB0}}$–$\overline{\text{DB7}}$, $\overline{\text{DBP}}$	40	ns
$\overline{\text{RST}}$	$\overline{\text{RSTOUT}}$	60	ns
$\overline{\text{ATN}}$	ATN/MODE1	80	ns
$\overline{\text{I}}/\overline{\text{O}}$	I/O_OUT/MODE2	80	ns

NOTE: All input timing is referenced to 0.8 V and 2.0 V. All output timing is referenced to 0.8 V and 2.0 V, except open-drain outputs, which are referenced to 0.5 V and 5.0 V, with the test loads shown in Figure 2.

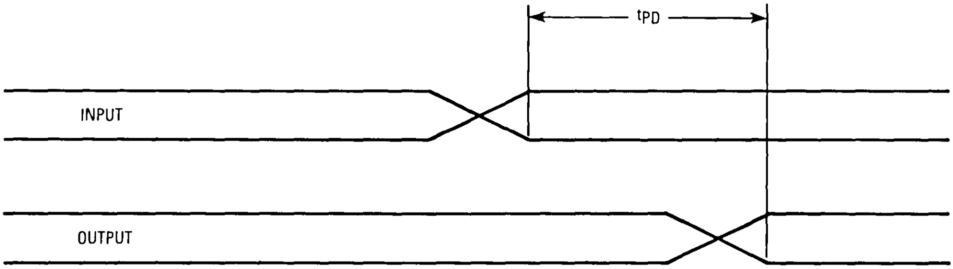
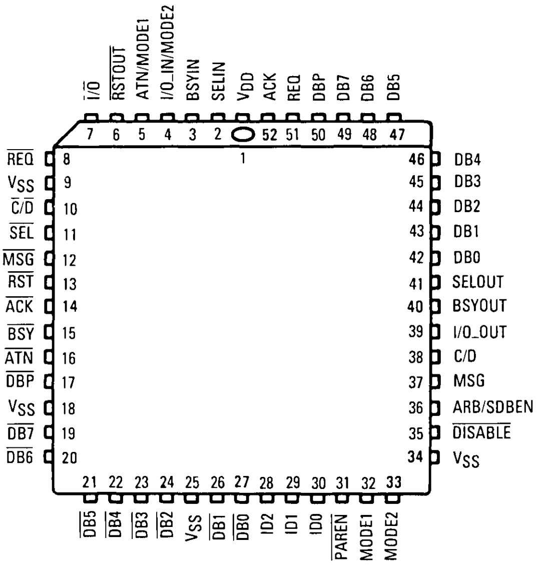


Figure 3. Transition Timing

ORDERING INFORMATION (T_A = 0–70°C)

Package Type	Order Number
PLCC FN Suffix	MC6898FN

PIN ASSIGNMENT



I/O
RSTOUT
ATN/MODE1
I/O_IN/MODE2
BSYIN
SELIN
VDD
ACK
REQ
DBP
DB7
DB6
DB5
7
6
5
4
3
2
52
51
50
49
48
47
1
REQ 8
VSS 9
C/D 10
SEL 11
MSG 12
RST 13
ACK 14
BSY 15
ATN 16
DBP 17
VSS 18
DB7 19
DB6 20
46 DB4
45 DB3
44 DB2
43 DB1
42 DB0
41 SELOUT
40 BSYOUT
39 I/O_OUT
38 C/D
37 MSG
36 ARB/SDBEN
35 DISABLE
34 VSS
21
22
23
24
25
26
27
28
29
30
31
32
33
DB5
DB4
DB3
DB2
VSS
DB1
DB0
ID2
ID1
ID0
PAREN
MODE1
MODE2

MC68HC99

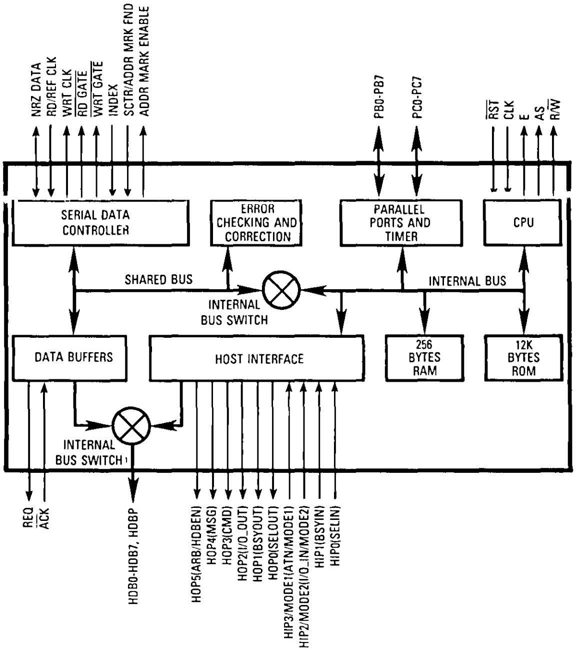
Technical Summary

Motorola's MC68HC99 hard disk controller (HDC) provides an economical solution to the problem of controlling one or more hard disks. The high-speed, low-cost HCMOS technology of the MC68HC99 replaces the multiple-chip set small computer system interface (SCSI) controller solutions used in current board-level and embedded controller designs. The following block diagram shows how the HDC incorporates all the elements of a disk controller in a single package.

Some hardware and software features of the MC68HC99 are as follows:

- High-Speed Serial Data Controller
- Low-Power HCMOS Technology
- Reed-Solomon Error Detection/Correction
- Two On-Chip 528-Byte Rotating Data Buffers
- MC68HC11 Central Processing Unit (CPU) with 256 Bytes of Random-Access Memory (RAM)
- MC68HC11 Core with 12K Bytes of Read-Only Memory (ROM)
- 16-Bit timer with Input Capture and Output Compare Functions
- Programmable Host Interface
  - Suitable for Both SCSI and Non-SCSI Applications
  - Completely SCSI Compatible
- Four Operating Modes
  - Single Chip
  - Simple Expanded (for External Peripheral Devices)
  - ROM Expanded (for Additional Firmware Space)
  - ROM Expanded with Timer (Adds External Timer Functions)

BLOCK DIAGRAM


NRZ DATA
RD/REF CLK
WRT CLK
$\overline{\text{RD GATE}}$
$\overline{\text{WRT GATE}}$
INDEX
SCTR/ADDR MRK FND
ADDR MARK ENABLE
PB0-PB7
PC0-PC7
$\overline{\text{RST}}$
CLK
E
AS
R/$\overline{\text{W}}$
SERIAL DATA CONTROLLER
ERROR CHECKING AND CORRECTION
PARALLEL PORTS AND TIMER
CPU
SHARED BUS
INTERNAL BUS
INTERNAL BUS SWITCH
DATA BUFFERS
HOST INTERFACE
256 BYTES RAM
12K BYTES ROM
INTERNAL BUS SWITCH
REQ
$\overline{\text{ACK}}$
HDB0-HDB7, HDBP
HOP5(ARB/HDBEN)
HOP4(MSG)
HOP3(CMD)
HOP2(I/O_OUT)
HOP1(BSYOUT)
HOP0(SELOUT)
HIP3/MODE1(ATN/MODE1)
HIP2/MODE2(I/O_IN/MODE2)
HIP1(BSYIN)
HIP0(SELIN)

This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the HDC using these two pins. V_{DD} is the power input (+5.0 V), and V_{SS} is ground.

CLOCK INPUT (CLK)

Clock input must be a transistor-to-transistor logic (TTL) compatible signal that is internally buffered to develop the internal clocks needed by the HDC. This clock input should not be gated off at any time.

E-CLOCK (E)

This clock output signal provides a timing reference for CPU bus cycles on the expanded bus. When E is low, an internal cycle is taking place. High E indicates that an external bus access is taking place. E is valid regardless of which mode the HDC is configured to operate in.

RESET ($\overline{\text{RST}}$)

This input signal is used to reset the HDC. A logic low on this pin preemptively resets the HDC and drives all outputs to their quiescent state.

ADDRESS STROBE (AS)

This output signal is used to demultiplex data signals from ports B and C. AS is high one-eight of a cycle after the falling edge of E, and low one-eight of a cycle before the rising edge of E. AS is used to capture low-order address information with a transparent latch. AS is valid regardless of which mode the HDC is configured to operate in.

READ/WRITE (R/$\overline{\text{W}}$

This output signal controls the direction of transfers on the external data bus. A high on this pin indicates a read cycle; a low indicates a write cycle. R/$\overline{\text{W}}$ is valid regardless of which mode the HDC is configured to operate in.

HOST DATA BUS (HDB0-HDB7, HDBP)

The host data bus consists of eight data signals (HDB0-HDB7), and a parity signal (HDBP). This bus is used to transfer commands, status, and data between the HDC and its host computers. Internally, the host data bus connects with the host data buffers and with the host data register (HDR).

HOST-INTERFACE INPUT/OUTPUT PINS (HIP0-HIP3, HOP0-HOP5)

The HDC host interface includes four input signals (HIP0-HIP3) and six output signals (HOP0-HOP5). Each input signal has a corresponding bit in the host status register (HSR). Each output signal has a corresponding bit in the host pin control register (HPCR). When the HDC is used as an SCSI controller, the output signals drive the select (SEL), busy (BSY), input/output (I/O), command (CMD), and message (MSG) signals. Under the same conditions, the input signals monitor the SEL, BSY, I/O, and attention (ATN) signals.

TRANSFER REQUEST (REQ) AND ACKNOWLEDGE (ACK)

The output signal, REQ, and input signal, ACK, are used to transfer data asynchronously between the HDC data buffers and the host computer.

SERIAL INTERFACE SIGNALS

The following are the high-speed serial data controller signals. Three of these signals, NRZ DATA, WRT CLK, and ADDR MARK ENABLE must be of equal current and capacitance in order to maintain timing relationships.

Nonreturn To Zero Data (NRZ DATA)

This bidirectional signal transfers serial NRZ data between the HDC and the disk.

Read/Reference Clock (RD/REF CLK)

This input signal from the disk determines the data transfer rate.

Write Clock (WRT CLK)

This output signal is derived from the reference clock input. It is supplied prior to beginning a write data operation and exists for the duration of the write operation.

Read Gate ($\overline{\text{RD}}$ $\overline{\text{GATE}}$)

When asserted, this active-low output signal indicates a disk read operation.

Write Gate ($\overline{\text{WRT}}$ $\overline{\text{GATE}}$)

When asserted, this active-low output signal indicates a disk write operation.

Index Detect (INDEX)

This input signal, when asserted, indicates the beginning of a track on the disk.

Sector/Address Mark Found (SCTR/ADDR MRK FND)

This input signal is dependent on the type of disk attached to the HDC. With a hard-sectored disk, this signal indicates the beginning of a sector. With a soft-sectored disk, this signal is driven by an external ADDRESS MARK DETECT circuit.

Address Mark Enable (ADDR MARK ENABLE)

The function of this input signal is dependent on the type of disk attached to the HDC and the programmed operation of the HDC.

If the HDC is programmed for the soft-sectored enhanced small disk interface (ESDI) disk format and the operation is write format, then this signal, asserted with $\overline{\text{WRT}}$ $\overline{\text{GATE}}$, causes the ESDI disk to write an address mark. Negation of ADDR MARK ENABLE with $\overline{\text{WRT}}$ $\overline{\text{GATE}}$ asserted indicates the beginning of an ID PLO sync field. In all functions other than write format, ADDR MARK ENABLE is asserted instructing the soft-sectored ESDI disk to search for an address mark field.

If the HDC is programmed for the hard-sectored ESDI disk format and the operation is write format, then ADDR MARK is asserted during the preamble field to indicate the beginning of an ID PLO sync field. ADDR MARK ENABLE is not asserted in operations other than write format.

If the HDC is programmed for the ST-506 disk format and the operation is write format, then ADDR MARK ENABLE is asserted with $\overline{\text{WRT GATE}}$ for one bit time during the sixth bit of the address mark field. This pulse is used externally to generate an illegal clock pattern that constitutes a unique address mark.

INPUT/OUTPUT PROGRAMMING

Three of the five I/O parallel port registers available with the MC68HC99 HDC are used in the serial data controller and the host interface. The function of the remaining parallel ports, B and C, is dependent on the operating mode selected.

SINGLE-CHIP MODE

The single-chip mode allows only the on-chip resource shown in the block diagram to be available to the HDC. The CPU fetches interrupt vectors and executes its code from the ROM.

In this mode, port B pins 0-7 are general-purpose I/O pins. Pins PB6 and PB7 can be programmed to function as the timer capture input signal (TCAP) and the timer compare output signal (TCMP), respectively. This reprogramming is accomplished by setting the TCAP/TCMP enable (TEN) bit of the highest priority interrupt register (HPIR).

All port C pins are general-purpose I/O pins in this mode.

EXPANDED MODE 1

Expanded mode 1 uses the internal ROM for user firmware and CPU fetches of all interrupt vectors. Additional peripheral devices or memory can be added via the expanded bus.

In this mode, all port B pins (0-7) act as high-order address output pins. During each MCU cycle, bits 8-15 of the address are output on PB0-PB7.

All port C pins are configured as multiplexed address/data pins. During the address portion of each microcomputer unit (MCU) cycle, bits 0-7 of the address are output on PC0-PC7. During the data portion of the MCU cycle (E high), bits PC0-PC7 are bidirectional data pins controlled by the R/$\overline{\text{W}}$ signal.

EXPANDED MODE 2

Expanded mode 2 uses external memory for CPU fetches of firmware and interrupt vectors. After reset, the internal ROM is left in the memory map. This ROM can be removed from the memory map by setting the remove internal ROM (REMROM) bit of the HPIR. This removal adds 12K to the external memory space.

All of the port B pins (0-7) act as high-order address output pins. During each MCU cycle, bits 8-15 of the address are output on PB0-PB7.

All port C pins are configured as multiplexed address/data pins. During the address portion of each MCU cycle, bits 0-7 of the address, are output on PC0-PC7. During the data portion of the MCU cycle (E high), bits PC0-PC7 are bidirectional data pins controlled by the R/$\overline{\text{W}}$ signal.

EXPANDED MODE 3

Expanded mode 3 is the same as expanded mode 2, except for the configuration of parallel port B.

Port B pins 0-5 act as high-order address output pins. During each MCU cycle, bits 8-13 of the address, are output on PB0-PB5. PB6 and PB7 are configured as multiplexed, high-order address/timer pins. During the address portion of each MCU cycle, bits 14 and 15 of the address are output on PB6 and PB7, respectively. During the data portion of the MCU cycle (E high), PB6 is TCAP (input), and PB7 is TCMP (output) for the timer.

All port C pins function the same way as in expanded mode 2.

NOTE

When the HDC is configured to operate in any of the expanded modes, data written to the internal registers and internal RAM is also driven onto the external data bus. If data is written to the internal ROM while the REMROM bit in the HPIR is logic zero, the data is not driven onto the external bus.

MEMORY

The MC68HC99 HDC has 12K bytes of available ROM and 256K bytes of available RAM. The memory is mapped for the HDC single-chip mode as shown in Figure 1, for the expanded mode 1 as shown in Figure 2, and for expanded modes 2 and 3 as shown in Figure 3.

REGISTERS

The CPU of the HDC uses the basic core of the MC68HC11 microcomputer. Seven CPU registers are available to programmers, see Figure 4.

ACCUMULATORS A AND B

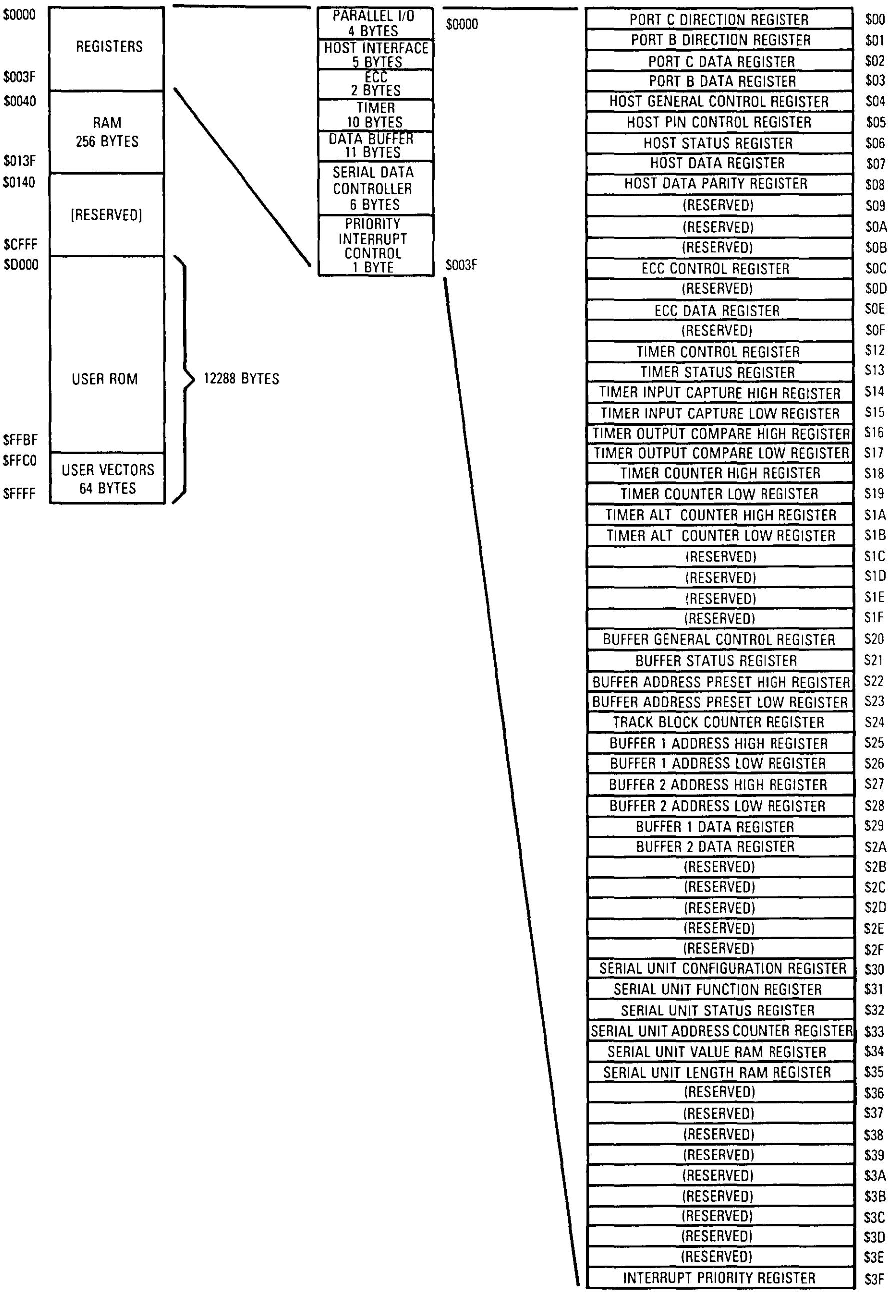
Accumulators A and B are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. The two accumulators can be concatenated into a single 16-bit accumulator called the D accumulator.

INDIRECT REGISTER X (IX)

The 16-bit IX register is used for indexed mode addressing. It provides a 16-bit indexing value that is added to an 8-bit offset in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

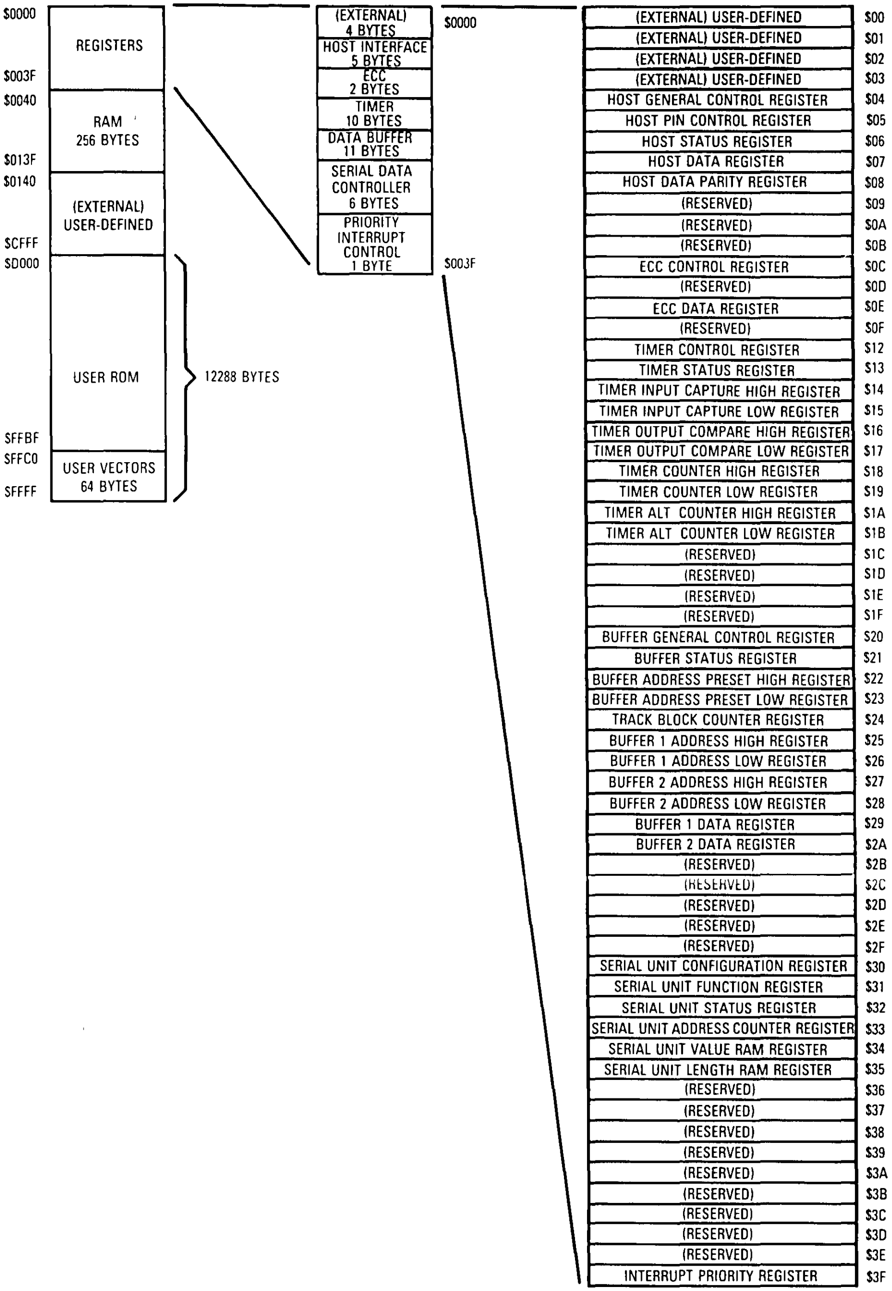
INDIRECT REGISTER Y (IY)

The 16-bit IY register is also used for indexed mode addressing, similar to the IX register. All instructions using the IY register require an extra byte of execution time because these instructions use off-page addressing.


$0000
REGISTERS
$003F
$0040
RAM
256 BYTES
$013F
$0140
(RESERVED)
$CFFF
$D000
USER ROM
12288 BYTES
$FFBF
$FFC0
USER VECTORS
64 BYTES
$FFFF
PARALLEL I/O
4 BYTES
HOST INTERFACE
5 BYTES
ECC
2 BYTES
TIMER
10 BYTES
DATA BUFFER
11 BYTES
SERIAL DATA
CONTROLLER
6 BYTES
PRIORITY
INTERRUPT
CONTROL
1 BYTE
$0000
$003F

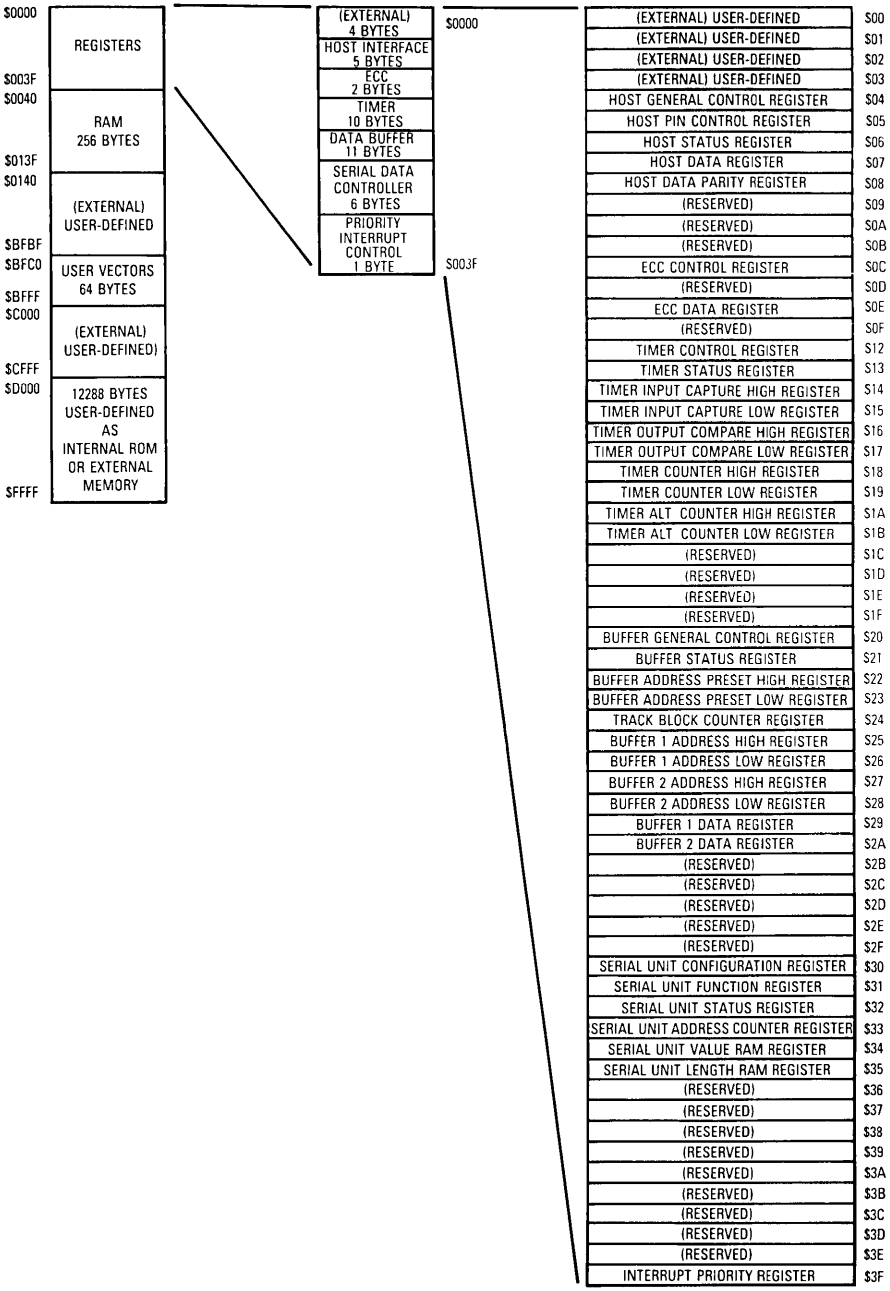
PORT C DIRECTION REGISTER	$00
PORT B DIRECTION REGISTER	$01
PORT C DATA REGISTER	$02
PORT B DATA REGISTER	$03
HOST GENERAL CONTROL REGISTER	$04
HOST PIN CONTROL REGISTER	$05
HOST STATUS REGISTER	$06
HOST DATA REGISTER	$07
HOST DATA PARITY REGISTER	$08
(RESERVED)	$09
(RESERVED)	$0A
(RESERVED)	$0B
ECC CONTROL REGISTER	$0C
(RESERVED)	$0D
ECC DATA REGISTER	$0E
(RESERVED)	$0F
TIMER CONTROL REGISTER	$12
TIMER STATUS REGISTER	$13
TIMER INPUT CAPTURE HIGH REGISTER	$14
TIMER INPUT CAPTURE LOW REGISTER	$15
TIMER OUTPUT COMPARE HIGH REGISTER	$16
TIMER OUTPUT COMPARE LOW REGISTER	$17
TIMER COUNTER HIGH REGISTER	$18
TIMER COUNTER LOW REGISTER	$19
TIMER ALT COUNTER HIGH REGISTER	$1A
TIMER ALT COUNTER LOW REGISTER	$1B
(RESERVED)	$1C
(RESERVED)	$1D
(RESERVED)	$1E
(RESERVED)	$1F
BUFFER GENERAL CONTROL REGISTER	$20
BUFFER STATUS REGISTER	$21
BUFFER ADDRESS PRESET HIGH REGISTER	$22
BUFFER ADDRESS PRESET LOW REGISTER	$23
TRACK BLOCK COUNTER REGISTER	$24
BUFFER 1 ADDRESS HIGH REGISTER	$25
BUFFER 1 ADDRESS LOW REGISTER	$26
BUFFER 2 ADDRESS HIGH REGISTER	$27
BUFFER 2 ADDRESS LOW REGISTER	$28
BUFFER 1 DATA REGISTER	$29
BUFFER 2 DATA REGISTER	$2A
(RESERVED)	$2B
(RESERVED)	$2C
(RESERVED)	$2D
(RESERVED)	$2E
(RESERVED)	$2F
SERIAL UNIT CONFIGURATION REGISTER	$30
SERIAL UNIT FUNCTION REGISTER	$31
SERIAL UNIT STATUS REGISTER	$32
SERIAL UNIT ADDRESS COUNTER REGISTER	$33
SERIAL UNIT VALUE RAM REGISTER	$34
SERIAL UNIT LENGTH RAM REGISTER	$35
(RESERVED)	$36
(RESERVED)	$37
(RESERVED)	$38
(RESERVED)	$39
(RESERVED)	$3A
(RESERVED)	$3B
(RESERVED)	$3C
(RESERVED)	$3D
(RESERVED)	$3E
INTERRUPT PRIORITY REGISTER	$3F

Figure 1. HDC Single-Chip Mode Memory Map


$0000
REGISTERS
$003F
$0040
RAM
256 BYTES
$013F
$0140
(EXTERNAL)
USER-DEFINED
$CFFF
$D000
USER ROM
12288 BYTES
$FFBF
$FFC0
USER VECTORS
64 BYTES
$FFFF
(EXTERNAL)
4 BYTES
HOST INTERFACE
5 BYTES
ECC
2 BYTES
TIMER
10 BYTES
DATA BUFFER
11 BYTES
SERIAL DATA
CONTROLLER
6 BYTES
PRIORITY
INTERRUPT
CONTROL
1 BYTE
$0000
$003F

(EXTERNAL) USER-DEFINED	$00
(EXTERNAL) USER-DEFINED	$01
(EXTERNAL) USER-DEFINED	$02
(EXTERNAL) USER-DEFINED	$03
HOST GENERAL CONTROL REGISTER	$04
HOST PIN CONTROL REGISTER	$05
HOST STATUS REGISTER	$06
HOST DATA REGISTER	$07
HOST DATA PARITY REGISTER	$08
(RESERVED)	$09
(RESERVED)	$0A
(RESERVED)	$0B
ECC CONTROL REGISTER	$0C
(RESERVED)	$0D
ECC DATA REGISTER	$0E
(RESERVED)	$0F
TIMER CONTROL REGISTER	$12
TIMER STATUS REGISTER	$13
TIMER INPUT CAPTURE HIGH REGISTER	$14
TIMER INPUT CAPTURE LOW REGISTER	$15
TIMER OUTPUT COMPARE HIGH REGISTER	$16
TIMER OUTPUT COMPARE LOW REGISTER	$17
TIMER COUNTER HIGH REGISTER	$18
TIMER COUNTER LOW REGISTER	$19
TIMER ALT COUNTER HIGH REGISTER	$1A
TIMER ALT COUNTER LOW REGISTER	$1B
(RESERVED)	$1C
(RESERVED)	$1D
(RESERVED)	$1E
(RESERVED)	$1F
BUFFER GENERAL CONTROL REGISTER	$20
BUFFER STATUS REGISTER	$21
BUFFER ADDRESS PRESET HIGH REGISTER	$22
BUFFER ADDRESS PRESET LOW REGISTER	$23
TRACK BLOCK COUNTER REGISTER	$24
BUFFER 1 ADDRESS HIGH REGISTER	$25
BUFFER 1 ADDRESS LOW REGISTER	$26
BUFFER 2 ADDRESS HIGH REGISTER	$27
BUFFER 2 ADDRESS LOW REGISTER	$28
BUFFER 1 DATA REGISTER	$29
BUFFER 2 DATA REGISTER	$2A
(RESERVED)	$2B
(RESERVED)	$2C
(RESERVED)	$2D
(RESERVED)	$2E
(RESERVED)	$2F
SERIAL UNIT CONFIGURATION REGISTER	$30
SERIAL UNIT FUNCTION REGISTER	$31
SERIAL UNIT STATUS REGISTER	$32
SERIAL UNIT ADDRESS COUNTER REGISTER	$33
SERIAL UNIT VALUE RAM REGISTER	$34
SERIAL UNIT LENGTH RAM REGISTER	$35
(RESERVED)	$36
(RESERVED)	$37
(RESERVED)	$38
(RESERVED)	$39
(RESERVED)	$3A
(RESERVED)	$3B
(RESERVED)	$3C
(RESERVED)	$3D
(RESERVED)	$3E
INTERRUPT PRIORITY REGISTER	$3F

Figure 2. HDC Expanded Mode 1 Memory Map



$0000	REGISTERS
$003F	
$0040	RAM 256 BYTES
$013F	
$0140	(EXTERNAL) USER-DEFINED
$BFBF	
$BFC0	USER VECTORS 64 BYTES
$BFFF	
$C000	(EXTERNAL) USER-DEFINED)
$CFFF	
$D000	12288 BYTES USER-DEFINED AS INTERNAL ROM OR EXTERNAL MEMORY
$FFFF	

$0000	(EXTERNAL) 4 BYTES
	HOST INTERFACE 5 BYTES
	ECC 2 BYTES
	TIMER 10 BYTES
	DATA BUFFER 11 BYTES
	SERIAL DATA CONTROLLER 6 BYTES
$003F	PRIORITY INTERRUPT CONTROL 1 BYTE

(EXTERNAL) USER-DEFINED	$00
(EXTERNAL) USER-DEFINED	$01
(EXTERNAL) USER-DEFINED	$02
(EXTERNAL) USER-DEFINED	$03
HOST GENERAL CONTROL REGISTER	$04
HOST PIN CONTROL REGISTER	$05
HOST STATUS REGISTER	$06
HOST DATA REGISTER	$07
HOST DATA PARITY REGISTER	$08
(RESERVED)	$09
(RESERVED)	$0A
(RESERVED)	$0B
ECC CONTROL REGISTER	$0C
(RESERVED)	$0D
ECC DATA REGISTER	$0E
(RESERVED)	$0F
TIMER CONTROL REGISTER	$12
TIMER STATUS REGISTER	$13
TIMER INPUT CAPTURE HIGH REGISTER	$14
TIMER INPUT CAPTURE LOW REGISTER	$15
TIMER OUTPUT COMPARE HIGH REGISTER	$16
TIMER OUTPUT COMPARE LOW REGISTER	$17
TIMER COUNTER HIGH REGISTER	$18
TIMER COUNTER LOW REGISTER	$19
TIMER ALT COUNTER HIGH REGISTER	$1A
TIMER ALT COUNTER LOW REGISTER	$1B
(RESERVED)	$1C
(RESERVED)	$1D
(RESERVED)	$1E
(RESERVED)	$1F
BUFFER GENERAL CONTROL REGISTER	$20
BUFFER STATUS REGISTER	$21
BUFFER ADDRESS PRESET HIGH REGISTER	$22
BUFFER ADDRESS PRESET LOW REGISTER	$23
TRACK BLOCK COUNTER REGISTER	$24
BUFFER 1 ADDRESS HIGH REGISTER	$25
BUFFER 1 ADDRESS LOW REGISTER	$26
BUFFER 2 ADDRESS HIGH REGISTER	$27
BUFFER 2 ADDRESS LOW REGISTER	$28
BUFFER 1 DATA REGISTER	$29
BUFFER 2 DATA REGISTER	$2A
(RESERVED)	$2B
(RESERVED)	$2C
(RESERVED)	$2D
(RESERVED)	$2E
(RESERVED)	$2F
SERIAL UNIT CONFIGURATION REGISTER	$30
SERIAL UNIT FUNCTION REGISTER	$31
SERIAL UNIT STATUS REGISTER	$32
SERIAL UNIT ADDRESS COUNTER REGISTER	$33
SERIAL UNIT VALUE RAM REGISTER	$34
SERIAL UNIT LENGTH RAM REGISTER	$35
(RESERVED)	$36
(RESERVED)	$37
(RESERVED)	$38
(RESERVED)	$39
(RESERVED)	$3A
(RESERVED)	$3B
(RESERVED)	$3C
(RESERVED)	$3D
(RESERVED)	$3E
INTERRUPT PRIORITY REGISTER	$3F

Figure 3. HDC Expanded Modes 2 and 3 Memory Map

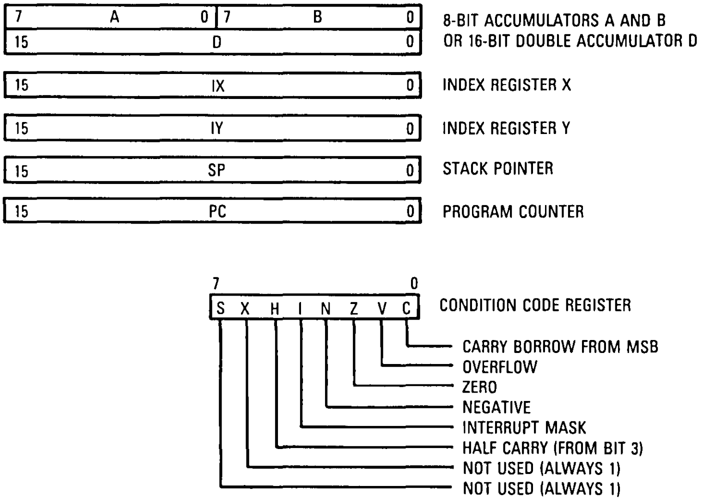

7
A
0
7
B
0
15
D
0
8-BIT ACCUMULATORS A AND B
OR 16-BIT DOUBLE ACCUMULATOR D
15
IX
0
INDEX REGISTER X
15
IY
0
INDEX REGISTER Y
15
SP
0
STACK POINTER
15
PC
0
PROGRAM COUNTER
7
0
S X H I N Z V C
CONDITION CODE REGISTER
CARRY BORROW FROM MSB
OVERFLOW
ZERO
NEGATIVE
INTERRUPT MASK
HALF CARRY (FROM BIT 3)
NOT USED (ALWAYS 1)
NOT USED (ALWAYS 1)

Figure 4. Programming Model

PROGRAM COUNTER (PC)

The PC is a 16-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

The SP is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of LIFO read/write registers that allow important data, such as the PC and CCR, to be stored during interrupts and subroutine calls. Each push decrements the stack; each pull increments it. Stacking order is shown in Figure 5.

CONDITION CODE REGISTER (CCR)

The CCR is an 8-bit register with each bit signifying the results of the instruction just executed. These bits are program testable, allowing specific action to be taken based on test results. The condition code register bits are given following.

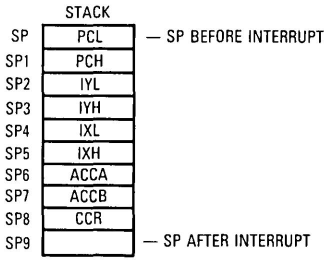

STACK
SP
PCL
— SP BEFORE INTERRUPT
SP1
PCH
SP2
IYL
SP3
IYH
SP4
IXL
SP5
IXH
SP6
ACCA
SP7
ACCB
SP8
CCR
SP9
— SP AFTER INTERRUPT

Figure 5. Interrupt Stacking Order

Carry/Borrow (C).

The C bit is set if a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last operation. The C bit is also effected during shift and rotate instructions.

Overflow (V)

The V bit is set if an arithmetic overflow resulted from the last operation. Otherwise, the V bit is cleared.

Zero (Z)

The Z bit is set if the result of the last arithmetic, logic, or data operation was zero. Otherwise, the Z bit is cleared.

Negative (N)

The N bit is set if the result of the last arithmetic, logic, or data operation was negative. Otherwise, the N bit is cleared.

INTERRUPT MASK (I)

The I bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources, (both external and internal).

Half Carry (H)

The H bit is set when a carry occurs between bits three and four of the ALU during an ADD, ABA, or ADC instruction. Otherwise, the H bit is cleared.

HOST INTERFACE

The host interface provides the connection between the HDC and one or more host computers. Although the host interface conforms to the proposed ANSI X3T9.2 SCSI bus specification, it can operate with other interfaces. The HDC host interface consists of five registers through which the firmware is able to:

- Control the host-interface operations,
- Monitor and manipulate the host-interface control signals, and
- Read or write the host-interface data bus.

In addition to these registers, the host interface contains special-purpose hardware that allows the HDC to function as an SCSI target device.

NOTE

> As a convention, reset values for register bits in all MC68HC99 registers are defined as follows:
> 0 = Bit is reset to zero.
> 1 = Bit is reset to one.
> U = Bit is not affected by reset.
> R = Bit value is determined by an input pin at reset.

HOST GENERAL CONTROL REGISTER (HGCR) $0004

The HGCR controls the host-interface interrupt activity. Bits HGCR5-HGCR7 are used only when the HDC is functioning as an SCSI controller.

	7	6	5	4	3	2	1	0
	REC	ARBINT	SELINT	HSTINT	0	0	0	0
				(ATNINT)				
RESET.	0	0	0	0	0	0	0	0

REC — Start Arbitration/Reconnection Operation Bit
When the operation is completed, REC is cleared by the host interface and cannot be reset to zero by a CPU write to the HGCR. REC should be used only when the HDC is an SCSI controller.
1 = Operation in progress
0 = Operation completed

ARBINT — Arbitration Begun Interrupt Enable Bit
ARBINT should be used only when the HDC is an SCSI controller.
1 = Enable
0 = Disable

SELINT — Selection Interrupt Enable Bit
SELINT should be used only when the HDC is an SCSI controller.
1 = Enable
0 = Disable

HSTINT — Host Interrupt Enable (Attention Interrupt Enable) Bit
1 = Enable
0 = Disable

Bits 3–0 — Not used.
Unused bits in the HGCR always read as zeros.

HOST PIN CONTROL REGISTER (HPCR) $0005

The HPCR controls the output levels of the host output port pins (HOP0–HOP5) as well as the direction of bits 0-7 of the host-interface data register and the data bus parity (DBP) bit in the host-interface data parity register.

	7	6	5	4	3	2	1	0
	DDIR	0	HOP5	HOP4	HOP3	HOP2	HOP1	HOP0
			(ARBEN/ HDBEN)	(MSG)	(CMD)	(I/O OUT)	(BSYOUT)	(SELOUT)
RESET	0	0	0	0	0	0	0	0

DDIR — Data Register Direction Bit
1 = The information in bits 0-7 of the host-interface data register and bit DBP of the host-interface data parity register is driven onto the HDB0–HDB7 and HDBP pins, respectively.
0 = The information in bits 0-7 of the host-interface data register and bit DBP of the host-interface data parity register reflects the levels on the HDB0–HDB7 and HDBP pins, respectively.

Bit 6 — Not Used.
Unused bits in the HPCR always read as zeros.

HOP5 — Host-Interface Output Control Bit 5
1 = Pin HOP5(ARB/HDBEN) is forced to a logic high level.
0 = Pin HOP5(ARB/HDBEN) is forced to a logic low level.

HOP4 — Host-Interface Output Control Bit 4
1 = Pin HOP4(MSG) is forced to a logic high level.
0 = Pin HOP4(MSG) is forced to a logic low level.
HOP3 — Host-Interface Output Control Bit 3
1 = Pin HOP3(CMD) is forced to a logic high level.
0 = Pin HOP3(CMD) is forced to a logic low level.
HOP2 — Host-Interface Output Control Bit 2
1 = Pin HOP2(I/O_OUT) is forced to a logic high level.
0 = Pin HOP2(I/O_OUT) is forced to a logic low level.
HOP1 — Host-Interface Output Control Bit 1
1 = Pin HOP1(BSYOUT) is forced to a logic high level.
0 = Pin HOP1(BSYOUT) is forced to a logic low level.
HOP0 — Host-Interface Output Control Bit 0
1 = Pin HOP0(SELOUT) is forced to a logic high level.
0 = Pin HOP0(SELOUT) is forced to a logic low level.

HOST STATUS REGISTER (HSR) $0006

The HSR provides the status of the host input port pins (HIP0-HIP3).

7	6	5	4	3	2	1	0
LOST	ARB	SELPH	HIP3E	HIP3	HIP2	HIP1	HIP0
			(ATNE)	(ATN)	(I/O_IN)	(BSYIN)	(SELIN)
RESET							
0	0	0	0	U	U	U	U

LOST — Arbitration Lost Status Bit
This bit denotes the failure of a host-interface arbitration and reselection operation due to successful arbitration by a faster or higher priority SCSI device.

LOST should be used only when the HDC is an SCSI controller.
1 = Arbitration loss occured.
0 = Arbitration loss did not occur.
ARB — Arbitration Begun Bit
ARB should be used only when the HDC is an SCSI controller.
1 = Detection of an SCSI bus free phase by the host-interface arbitration and reselection operation drives the ARB/SDBEN and BSYOUT pins to logic high levels.
0 = Bus free phase not detected.
SELPH = Selection Phase Detected Bit
SELPH should be used only when the HDC is an SCSI controller.
1 = Host interface detects an SCSI selection phase.
0 = No SCSI selection phase detected.
HIP3E — Host Input Port Bit 3 Edge Detected Bit
1 = Host interface hardware detects a rising edge on the HIP3/MODE1(ATN/MODE1) pin.
0 = No rising edge detected on the HIP3/MODE1(ATN/MODE1) pin.
HIP3 — Host Input Port Bit 3
HIP3 (ATN) is a read-only status bit.
1 = Pin HIP3/MODE1 (ATN/MODE1) is at a logic high level.
0 = Pin HIP3/MODE1 (ATN/MODE1) is at a logic low level.
HIP2 — Host Input Port Bit 2
HIP2(I/O IN) is a read-only status bit.
1 = Pin HIP2/MODE2(I/O IN/MODE2) is at a logic high level.
0 = Pin HIP2/MODE2(I/O IN/MODE2) is at a logic low level.
HIP1 — Host Input Port Bit 1
HIP1(BSYIN) is a read-only status bit.
1 = Pin HIP1(BSYIN) is at a logic high level.
0 = Pin HIP1(BSYIN) is at a logic low level.
HIP0 — Host Input Port Bit 0
HIP0(SELIN) is a read-only status bit.
1 = Pin HIP0(SELIN) is at a logic high level.
0 = Pin HIP0(SELIN) is at a logic low level.

The LOST, ARB, SELPH, and HIP3E(ATNE) bits are cleared by a read of the HSR followed by a write of zero to the desired bit(s).

HOST DATA REGISTER (HDR) $0007

The HDR provides the means to place data on the HDB0–HDB7 pins and to read data off the HDB0–HDB7 pins. The data direction bit (DDIR) in the host pin control register (HPCR) determines whether the data in the HDR is driven on the HDB0–HDB7 pins or whether the data in the HDR reflects the levels on the HDB0–HDB7 pins. Bit 7 in the HDR corresponds to HDB7 and so on. A one in any bit indicates the level on the corresponding pin is a logic high, while a zero in any bit indicates the level on the corresponding pin is a logic low.

HOST DATA PARITY REGISTER (HDPR) $0008

The HDPR provides the means to place the correct parity on the HDBP pin in conjunction with the data in the HDR. Likewise, it can be used to monitor the level on the HDBP pin. It also provides information on whether or not parity is used on the host data bus, and it can store an encoded device ID.

7	6	5	4	3	2	1	0
DBP	HDBPEN	0	0	0	ID2	ID1	ID0
RESET.							
U	HDBP	0	0	0	HDB2	HDB1	HDB0

DBP — Data Bus Parity Bit
The DDIR bit in the HPCR determines whether the logic level on the DBP bit is driven onto the HDBP pin or reflects the logic level on the HDBP pin.
1 = HDBP is high.
0 = HDBP is low.
HDBPEN — Host Data Bus Parity Option Bit
This read-only bit reflects the value of the HDBP pin at the rising edge of $\overline{\text{RESET}}$.
1 = Host data bus parity is enabled.
0 = Host data bus parity is disabled.
Bits 5–3 — Not used.
These bits always read as zero.
ID2–0 — SCSI Bus Device Identification Bits
These are read-only bits that reflect the state of the HDB2–0 pins at the rising edge of $\overline{\text{RESET}}$.

HDC SCSI OPERATIONS

The following paragraphs provide information on how to use the HDC host-interface as an SCSI target. The HDC host-interface is capable of disconnecting and reconnecting and of detecting the SCSI selection, bus free, and attention phases. It can also perform the initial steps of the SCSI bus arbitration and reselection phases. The host interface provides the means:

1) To determine the levels of the SCSI BSY, SEL, I/O, and ATN signals and
2) To drive the SCSI BSY, SEL, I/O, CMD, and MSG signals.

CONTROLLING THE SCSI BUS

Once selected by an SCSI host, the HDC, as an SCSI target, controls the various types of information transferred on the SCSI data bus. The SCSI standard supports six information phases as follows:

Data out (from the host)	Data in (to the host)
Command out (from the host)	Status in (to the host)
	Message in (to the host)
Message out (from the host)	

The HDC host interface selects the information phase with the arbitration enabled/host data bus enabled (ARB/HDBEN), CMD, MSG, and I/O OUT signals as shown in Table 1. When the HDC is not communicating with a host, the APB/HDBEN, CMD, MSG, and I/O_OUT signals should be driven to a logic low level. All HDC host-interface signals are active high and should be inverted by the external hardware that drives the SCSI bus.

Table 1. SCSI Bus Phase Control

ARB/HDBEN	CMD	MSG	I/O_OUT	SCSI BUS PHASE
0	0	0	0	Data Out
0	1	0	0	Command
0	1	1	0	Message Out
1	0	0	1	Data In
1	1	0	1	Status
1	1	1	1	Message In
1	0	0	0	*

NOTE:
1 = Logic High
0 = Logic Low

*When ARB/HDBEN is 1 and I/O_Out is 0, external hardware drives the HDC SCSI bus device ID onto the SCSI data bus.

All data is transferred between the HDC and the host using the HDC data buffers. Using the corresponding control bits in the HPCR, the ARB/HDBEN, CMD, MSG, and I/O OUT signals are driven to select the desired data phase. Then, the data buffers transfer the data.

SERIAL DATA CONTROLLER

The serial data controller provides a high-speed connection between the HDC and a disk drive. This controller is a programmable unit that can work with several driven interfaces, such as ST-506, ESDI, and storage module device (SMD) interfaces, without CPU intervention. Operations performed are read and write standard, read and write format, read and write long, and data search.

The serial data controller consists of six registers through which the CPU can control serial operations. Three of these registers program the serial data operation, while the other three establish format parameters.

SERIAL UNIT CONFIGURATION REGISTER (SUCR) $0030

The SUCR selects the disk interface type.

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	FIXED	ESDI
RESET	0	0	0	0	0	0	0	0

SUCR 7–2 — Not used.
Unused bits in the SUCR always read as zeros.

FIXED — Disk Sector Type Bit
The FIXED bit should not be set when the ESDI bit is cleared.
1 = Hard-sectored
0 = Soft-sectored

ESDI — Disk Data Format Type Bit
1 = ESDI compatible disk
0 = ST-506 compatible disk

SERIAL UNIT FUNCTION REGISTER $0031

The serial unit function register (SUFR) selects and starts a serial data operation.

	7	6	5	4	3	2	1	0
	START	ABORT	INT EN	ECC	DIS-CHCK	FOR-MAT	RD/WRT	SEARCH
RESET	0	0	0	0	0	0	0	0

START — Start Operation Bit
The START bit always reads as a zero.
1 = The serial data controller begins the operation defined by the ECC, DISCHCK, FORMAT, RD/WRT, and SEARCH bits.
0 = Immediately after START is set to one, the serial data controller resets START to zero.

ABORT — Abort Operation Bit
Usually, ABORT will read as a zero.
1 = The serial data controller aborts the current operation within one sector time cycle, and performs an orderly shutdown of all serial control signals.
0 = Immediately after the ABORT bit is set to one, the serial data controller resets it to zero. The time required by the controller to clear ABORT depends on the frequency of the read/reference clock signal from the disk.

INT EN — Interrupt Enable Bit
1 = The serial data controller interrupt is enabled.
0 = The serial data controller interrupt is disabled.

ECC — Error Checking and Correction Bit
1 = When this bit is set, the serial data controller performs the read or write operation without using the ECC unit. On a read operation, the data field is not checked for data errors. This function allows a host computer to manage the ECC.
0 = (Default) When this bit is clear, the serial data controller performs the read or write operation using the ECC unit. Check bytes from the ECC unit are written on the disk immediately following the data field when a write operation is executed. When a read operation is executed, the ECC unit checks for data errors.

DISCHCK — Disable Header ECC Check Bit
1 = The address field of each sector is not checked for errors.
0 = (Default) The address field of each sector is checked for errors.

FORMAT — Selects a Format Operation Bit
1 = Selects a format track operation
0 = Does not select a format track operation.

RD/WRT — Selects a Read or Write Operation Bit
1 = Selects a read operation
0 = Selects a write operation

SEARCH — Selects a Search Operation Bit
1 = Selects a data search operation
0 = Does not select a data search operation

The serial data controller supports eight legal combinations of the ECC, DISCHCK, FORMAT, RD/WRT, and SEARCH bits. If an illegal option is selected, the serial data controller will operate erratically and can destroy data on the disk. Table 2 is a list of legal values in the various types of serial data controller operations.

Table 2. List of Legal Values

Serial Data Controller Operation
0 0 0 0 0 = Write Standard
0 0 0 1 0 = Read Standard
0 0 1 0 0 = Write Format
0 0 1 1 0 = Read Format
1 0 0 0 0 = Write with ECC
1 0 0 1 0 = Read with ECC
0 0 0 1 1 = Search
1 1 0 1 0 = Read with ECC on Header Disabled
All Others = Illegal Operation

SERIAL UNIT STATUS REGISTER (SUSR) $0032

The SUSR indicates the status of the serial data operation.

7	6	5	4	3	2	1	0
ACT	DONE	NO-MATCH	SKERR	FLGERR	IXERR	ECCERR	DSNF
RESET:							
0	0	0	0	0	0	0	0

ACT — Active Bit
One of the other status bits indicates whether or not it was a normal operation termination. ACT is a read-only status bit.
1 = The serial data controller operation is in progress.
0 = The operation terminated.

DONE — Operation Complete Bit
DONE is not valid while ACT is one.
1 = The serial data controller operation completed without error or was aborted by the ABORT bit in the SUFR.
0 = Error termination.

NOMATCH — No Match Bit
NOMATCH is not valid while ACT is one.
1 = The data search operation terminated because of a data miscompare.
0 = Data miscompare did not cause error termination.

SKERR — Seek Error Bit
SKERR is not valid while ACT is one.
1 = The serial data controller operation terminated because of a cylinder or head miscompare during an address field search.
0 = A cylinder or head miscompare did not cause error termination.

FLGERR — Flag Error Bit
FLGERR is not valid while ACT is one.
1 = The operation terminated because of a flag miscompare detected on the current track.
0 = A flag miscompare did not cause error termination.

IXERR — Index Error Bit
IXERR is not valid while ACT is one.
1 = Two index pulses were detected since the operation started without finding the indicated sector address. During a write format operation, format information was still being written when INDEX was detected, indicating the field lengths in the length RAM are too long.
0 = Index error did not cause error termination.

ECCERR — Error Checking and Correction Error Bit
ECCERR is not valid while ACT is one.
1 = A data field ECC error occurred.
0 = A data field error did not cause an error termination.

DSNF — Data Sync Not Found Bit
DSNF is valid only for ST-506 disks. DSNF is not valid when ACT is one.
1 = A data sync byte was never found during the operation.
0 = Data sync not found did not cause an error termination.

The DONE, NOMATCH, SKERR, FLGERR, IXERR, ECCERR, and IDSNF bits are cleared by a read of the SUSR followed by a write of zero to the desired bit(s).

SERIAL UNIT ADDRESS COUNTER REGISTER (SUAC)

The SUAC is a 5-bit register that points to two bytes: one in the value RAM and one in the length RAM. When a serial data operation is in progress, the SUAC register is under the control of the serial data controller and should not be accessed.

SERIAL UNIT VALUE RAM DATA REGISTER (SUVR)

The SUVR accesses the data in the value RAM.

SERIAL UNIT LENGTH RAM DATA REGISTER (SULR)

The SULR accesses the data in the length RAM.

VALUE AND LENGTH RAM ARRAYS

The serial data controller uses two arrays of RAM to define the length and data content of each field in one sector. The first array is the length RAM. The second array is the value RAM. A byte in the length RAM corresponds to a byte in the value RAM. Each RAM array contains 24 bytes, which are not directly mapped in the CPU's address space. As a result, the length and value RAM arrays are accessed via the SULR and SUVR, respectively. Each register is a window providing access to the corresponding bytes addressed by the SUAC.

During a serial data controller write format operation, the information in a value RAM byte is duplicated on the disk the number of times specified by the corresponding length RAM byte. Normal serial-data-controller read/write operations use the information in the value and length RAMS to assert/negate the serial-data-controller signals ($\overline{\text{RD GATE}}$, $\overline{\text{WRT GATE}}$, etc.). The serial-data-controller takes these steps to locate the data fields within the sectors on a track.

The value and length RAM arrays must be initialized by the CPU in the HDC startup procedure. Once the arrays are initialized, only the cylinder, head, sector, and flag fields need to be changed prior to starting each serial data operation.

BUFFER UNIT

The data buffer controller provides a high-speed connection between the HDC and a host computer using the host interface data bus. In a standard read operation, parallel data from the serial data controller is shipped through the data buffers to the host computer. A standard write operation ships data from the host through the data buffer to the serial data controller. When performing a standard read or write operation, the data buffer controller fills one data buffer while the other is being emptied. The data buffers are also used to send or receive command and status information between the HDC and a host computer and to send format information and verification patterns to the disk.

The data buffer controller consists of eight registers, through which the CPU can control data operations and

	7	6	5	4	3	2	1	0
	START	ABORT	DNINT	ERRINT	DEFALT	XFER1	XFER0	DIR
RESET:	0	0	0	0	0	0	0	0

START — Start Operation Bit

The START bit always reads logic zero.

1 = The buffer controller begins the operation defined by the XFER1, XFER0, and DIR bits.

0 = Immediately after START is set to one, the data buffer controller resets START to zero.

ABORT — Abort Operation Bit

The ABORT bit always reads logic zero.

1 = The data buffer controller aborts the current operation as soon as the byte transfer to/from the host and/or the byte transfer to/from the serial data controller is finished. The data buffer controller negates REQ and drives HDB0–HDB7 and HDBP to a quiescent state.

0 = Immediately after ABORT is set to one, the data buffer controller resets ABORT to zero.

DNINT — Done Interrupt Bit

1 = The buffer done interrupt is enabled.

0 = The buffer done interrupt is disabled.

ERRINT — Error Interrupt Bit

1 = The buffer error interrupt is enabled.

0 = The buffer error interrupt is disabled.

DEFALT — Default Bit

The default buffer is the first buffer to receive or send data.

1 = The programmed operation uses BUFFER2.

0 = The programmed operation uses BUFFER1.

XFER0–XFER1 — Transfer Zero–Transfer One Bits

These bits contain a binary number that selects the data buffer controller operation.

DIR — Direction Bit

1 = Direction of data flow through the data buffer is from controller to host.

0 = Direction of data flow through the data buffer is from host to serial data controller.

In this register, the DEFALT, XFER1, XFER0, and DIR bits are write protected when the ACT bit of the BUSR is logic one. Table 3 lists the data buffer controller transfer types and their coding.

Table 3. Transfer Types and Coding

Transfer Type
0 0 0 = Write Disk
0 0 1 = Read Disk
0 1 0 = Read Host
0 1 1 = Write Host
1 0 0 = Write Format

	7	6	5	4	3	2	1	0
	ACT	DONE	HPRTY1	DPRTY1	HPRTY2	DPRTY2	CNTERR	BUFX
RESET:	0	0	0	0	0	0	0	0

ACT — Operation Active Bit
ACT is a read-only status bit.
1 = A data bufer controller operation is in progress.
0 = The operation has terminated.
The other BUSR status bits indicate whether the termination was a normal or error termination.

DONE — Operation Done Bit
DONE is not valid while ACT is one.
1 = The operation completed without error or was aborted by setting the ABORT bit in the BGCR.
0 = The operation completed with error.

HPRTY1 — Parity Error Between Host and BUFFER1 Bit
HPRTY1 is not valid while ACT is one.
1 = The data buffer operation on BUFFER1 terminated because of a parity error on the host-interface data bus. If parity is not enabled on the host-interface data bus, then HPRTY1 is set to one only for a parity error on data transferred from BUFFER1 to the host computer.
0 = No parity error existed on data transferred between BUFFER1 and the host.

DPRTY1 — Parity Error Between Disk and BUFFER1 Bit
DPRTY1 is not valid while ACT is one.
1 = The data buffer operation on BUFFER1 terminated because of a parity error on data transferred from BUFFER1 to the serial data controller.
0 = No parity error existed on data transferred between BUFFER1 and the serial data controller.

HPRTY2 — Parity Error Between Host and BUFFER2 Bit
HPRTY2 is not valid while ACT is one.
1 = The data buffer operation on BUFFER2 terminated because of a parity error on the host interface data bus. If parity is not enabled on the host-interface data bus, then HPRTY2 is set to one only for a parity error on data transferred from BUFFER2 to the host computer.
0 = No parity error existed on data transferred between BUFFER2 and the host.

DPRTY2 — Parity Error Between Disk and BUFFER2 Bit
DPRTY2 is not valid while ACT is one.
1 = The data buffer operation on BUFFER2 terminated because of a parity error on data transferred from BUFFER2 to the serial data controller.
0 = No parity error existed on data transferred between BUFFER2 and disk.

CNTERR — Count Error Bit
CNTERR is not valid while ACT is one.
1 = The data buffer controller sent or received more bytes than the serial data controller requested.
0 = No count error occurred.

BUFX — Buffer Last Connected to Disk Bit
This information is used when an error occurred during a standard operation, and the data is to be corrected.
BUFX is a read-only status bit.
1 = The last buffer to transfer data to or from the serial data controller was BUFFER 2.
0 = The last buffer to transfer data to or from the serial data controller was BUFFER 1.

The DONE, HPRTY1, DPRTY1, HPRTY2, DPRTY2, and CNTERR bits are cleared by a read of the BUSR followed by a write to the desired bit(s).

BUFFER ADDRESS REGISTERS (BAR1, BAR2)

The buffer address registers (BAR1 and BAR2) are 16-bit registers used to address a byte in the corresponding data bufer. BAR1 and BAR2 are decremented by the data buffer controller when the corresponding data buffer is accessed. When a data buffer operation is in progress, BAR1 and BAR2 are controlled by the data buffer controller and should not be accessed.

The four most significant bits of both BAR1 and BAR2 are not used and always read as zeros. The address of the most significant byte (MSB) of BAR1 is $0025, and the address of the least significant byte (LSB) of BAR1 is $0026. The address of the MSB of BAR2 is $0027, and the address of the LSB of BAR2 is $0028.

BUFFER DATA REGISTERS (BDR1, BDR2)

Data in BUFFER1 and BUFFER2 is accessed via BDR1 and BDR2, respectively. When a data buffer operation is in progress, BDR1 and BDR2 are controlled by the data buffer controller and should not be accessed. The address of BDR1 is $0029, and the address of BDR2 is $002A.

BUFFER ADDRESS PRESET REGISTER (BAPR)

The BAPR is a 16-bit register used to load a preset address into BAR1 and/or BAR2 during a data buffer operation. The value in this register addresses the first byte in either buffer to send or receive data. The BAPR must be initialized prior to starting the data buffer operation. For example, when a 10-byte command packet is sent from the host computer into BUFFER1, BAPR is set to 9. After the data buffer operation, the first byte of the command is in BUFFER1 at address $009, and the last byte is at address $000. For a multiblock disk read with a block size of 256 bytes, the BAPR is set to $0FF (255). When a data buffer operation is in progress, BAPR is controlled by the buffer data controller and should not be accessed. The four most significant bits of the BAPR are not used and always read as zeros. The address of the MSB of the BAPR is $0022, and the address of the LSB is $0023.

TRACK BLOCK COUNTER REGISTER (TBCR)

The TBCR specifies the number of blocks of data to be transferred through the data buffers. For example, when a command packet is to be received from the host, TBCR is set to one. When a multiblock disk read is to be done, TBCR is set to the number of blocks to be sent to the host. During a data buffer operation, TBCR is controlled by the data buffer controller and should not be accessed. The address of the TBCR is $0024.

DATA BUFFERS

The HDC has two data buffers, BUFFER1 and BUFFER2, which are each large enough to hold a 512-byte sector of data plus 16 ECC check byte (a total of 528 bytes). During a standard read or write data operation, only one sector of data is held in a data buffer, and the two data buffers cannot be concatenated for larger sector sizes. Because the data buffers are not directly mapped in the CPU's address space, each buffer is accessed via the buffer data registers (BDR1 and BDR2) and the buffer address registers (BAR1 and BAR2). To read or write in BUFFERx, where x is 1 or 2, BARx must be set to the address of the byte ($000 through $20F). BDRx must then be read or written to access the data byte. After BDRx is read or written, BARx is automatically decremented by one. Figure 6 is a diagram of the data buffer.

If BARx is set to a value greater than $20F and BDRx is read or written, then BUFFERx is not accessed. In the case of a read, the value returned is undefined. The flowchart in Figure 7 shows how to access a data buffer.

DATA BUFFER PARITY

Each byte in both data buffers has a corresponding parity bit. When a data byte is written to either data buffer, a parity bit is generated and stored with the byte. When a data byte is read from a buffer, the byte is checked for a parity error. If an error is detected on a byte being sent to a host computer, then either the HPRTY1 or HPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte. If an error is detected on a byte being sent to the serial data controller, then either the DPRTY1 or DPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte.

When a data byte is received from the host computer with parity on the host interface data bus enabled, the byte is checked for a parity error. If an error is detected, then either the HPRTY1 and HPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte.

When a data byte is received from the host computer with parity on the host interface data bus enabled, the byte is checked for a parity error. If an error is detected, then either the HPRTY1 and HPRTY2 bit in the BUSR is set, depending on which data buffer received the byte.

No indication of a parity error when the CPU reads data from the data buffers is given.

ECC UNIT

The serial data controller manages the error checking and correction unit (ECC) to provide Reed-Solomon based error checking and correction without host intervention. When an error is detected on a data field, the CPU interrogates the ECC unit for information to correct erroneous data held in the data buffers. The ECC unit uses interleaving to increase protection for data fields by breaking the data field into smaller logical blocks and protecting each block separately. The interleave factor is selectable between one, two, three, and five. A recommended interleave of two should be used for block sizes of 256 or 128, and an interleave of three is recommended for a block size of 512. The ECC unit interleave should not be confused with the sector interleave on the disk. The ECC unit interleave, which is invisible, does not affect the way data is written on the disk.

The ECC unit consists of two registers through which the CPU can select an interleave factor and obtain correction information in case of error.

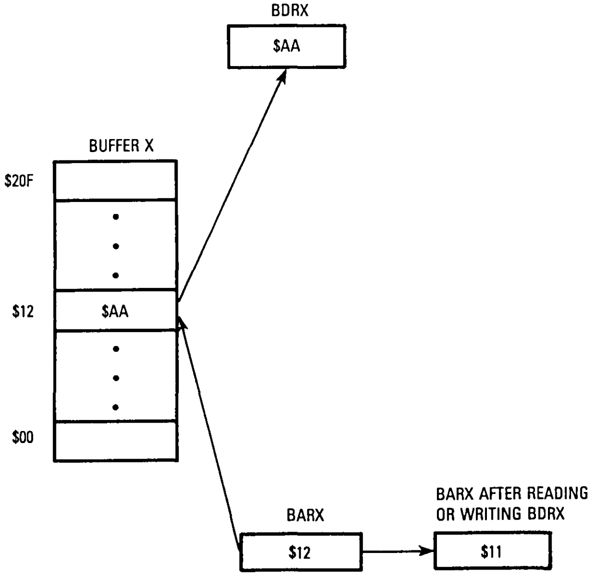
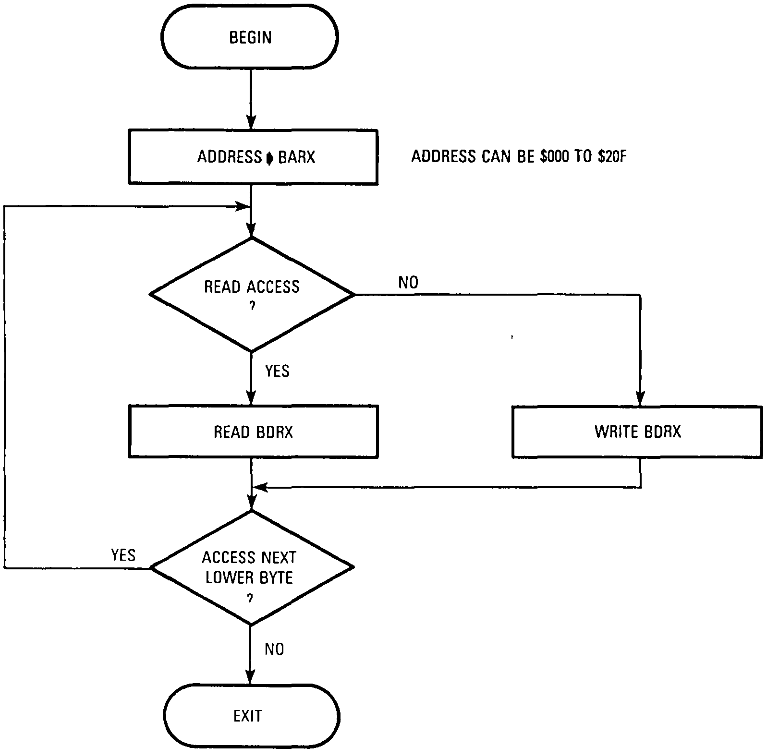


Figure 6. Data Buffer Diagram



AFTER THE ACCESS, BARX IS DECREMENTED BY ONE, SO THE NEXT LOWER BYTE IN THE BUFFER CAN BE ACCESSED WITHOUT WRITING TO BARX

Figure 7. MPU Access of Buffer X

ECC CONTROL REGISTER (ECR) $000C

The ECR selects the interleave and interrogates the ECC unit for correction information if an error is detected.

7	6	5	4	3	2	1	0
EM	IPT	ILV1	ILV0	CS	CP	CD	CR
RESET:							
U	U	U	U	U	U	U	U

EM — Enable Monitor Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

IP — Increment Pointer Bit
This bit is used during a correction to increment the interleave pointer.

ILV1–0 — Interleave Factor Bits
The ILV1 and ILV0 bits select the ECC interleave factor to be used on a block of data, as shown in Table 4.

Table 4. Interleave Selection

ILV1	ILV0	Interleave Factor
0	0	1
0	1	2
1	0	3
1	0	4

CS — Clear Syndrome Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

CP — Clear Pointer Bit
This bit is used during a correction to clear an internal data pointer.

CD — Clear Data Bit
This bit is used during a correction to clear the ECC data register.

CR — Clear RAM Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

ECC DATA REGISTER (EDCR)

The EDCR provides the CPU with an error byte mask and is used only during a correction.

RESET

The RESET signal resets the CPU core and on-chip devices to provide an orderly startup procedure and to ensure that all peripherals and output signals are in a quiescent state.

CPU AFTER RESET

After reset, if the operating mode is single-chip or expanded mode 1, the CPU fetches its restart vector from internal ROM at $FFFE and $FFFF. The I bit in the condition code register is set to mask interrupt requests.

PORTS B AND C AFTER RESET

After reset, if the operating mode is single-chip, then the port B and C pins are inputs. If the operating mode is one of the expanded modes, then the port B and C pins begin driving the expansion bus.

TIMER AFTER RESET

After reset, the timer counter is initialized to $FFF9. If the operating mode is single-chip, then timer TCAP and TCMP pin functions are disabled. The user can elect to enable these functions by setting the TEN bit of the HPIR.

INTERRUPTS

The HDC supports both maskable and nonmaskable interrupts with a prioritization scheme similar to the MC68HC11.

Maskable interrupts originate only from the on-chip peripheral devices and can be masked by the I bit of the condition code register. The interrupt signals coming from the various devices are prioritized. The highest priority interrupt is selected by changing the value of the PSEL0–3 bits in the highest priority interrupt register (HPIR). The timer interrupt defaults to the highest priority after reset.

Nonmaskable interrupts include RESET, illegal opcode trap, and software interrupt (SWI). RESET has the highest priority, followed by illegal opcode trap. SWI is actually an instruction having the highest priority after RESET. This statement is essentially true since, once the SWI opcode has been fetched, no other interrupt will be honored until after the SWI vector is fetched.

Table 5 shows the interrupt sources and their corresponding vectors. Each vector requires two bytes. The first vector byte contains the eight most significant bits of vector address; the second byte contains the eight least significant bits of vector address. Figure 8 shows the interrupt handler flowchart.

Table 5. Interrupt Vectors

Vector Address	Interrupt Source	Masked By
XXC0, XXC1 To XXE4, XXE5	Reserved	
XXE6, XXE7	Serial Unit Termination	I Bit
XXE8, XXE9	Buffer Done	I Bit
XXEA, XXEB	Buffer Error	I Bit
XXEC, XXED	SCSI Attention	I Bit
XXEE, XXEF	SCSI Arbitration Begun	I Bit
XXF0, XXF1	SCSI Selection Phase	I Bit
XXF2, XXF3	Timer	I Bit
XXF4, XXF5	Reserved	
XXF6, XXF7	SWI	None
XXF8, XXF9	Illegal Opcode Trap	None
XXFA, XXFB	Reserved	
XXFC, XXFD	Reserved	
XXFE, XXFF	Reset	None

NOTE:
XX = FF if mode is single-chip or expanded 1
XX = BF if mode is expanded 2 or expanded 3

HIGHEST PRIORITY INTERRUPT REGISTER (HPIR) $003F

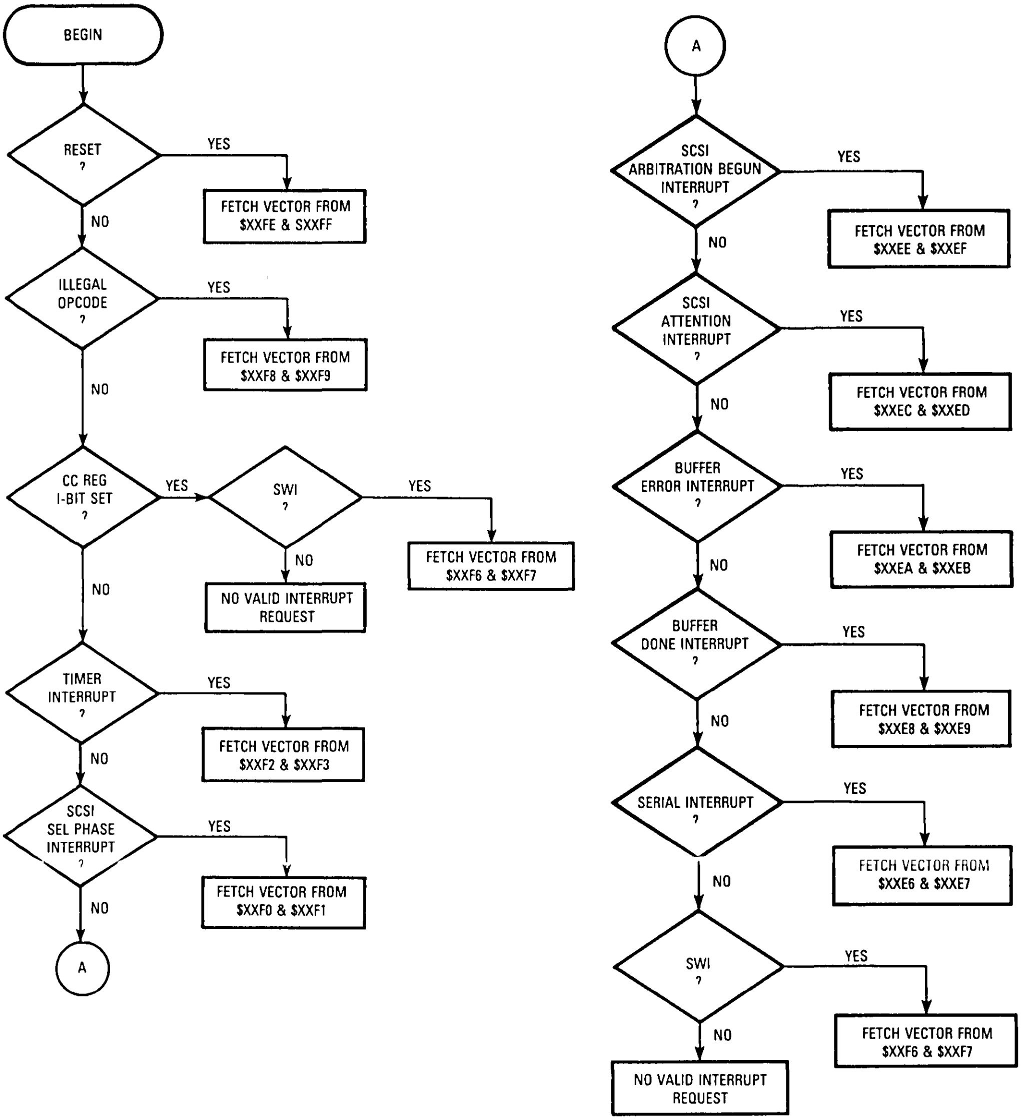
The HPIR is an 8-bit read/write register that contains seven control bits and one read-only status bit.

	7	6	5	4	3	2	1	0
	TEN	REM-ROM	IRV	MODE	PSEL3	PSEL2	PSEL1	PSEL0
RESET.	0	0	0	R	0	1	0	1

TEN — TCAP/TCMP Enable Bit
In single-chip mode:
1 = TCAP and TCMP are enabled on port B pins 6 and 7, respectively.
0 = TCAP and TCMP are disabled, and port B pins 6 and 7 are dedicated to general-purpose I/O.
In expanded modes:
TEN is a read-only bit that always reads logic zero.

REMROM — Remove Internal ROM Bit
In expanded 2 and 3 modes:
1 = Internal ROM is disabled, addresses from $D000 to $FFFF must be accessed externally.
0 = Internal ROM is enabled.
In expanded 1 and single-chip modes:
REMROM is a read-only bit that always reads logic zero.

IRV — Internal Read View Bit
In single-chip mode:
IRV is a read-only bit that always reads logic zero.
In expanded modes:
Once set, IRV can only be cleared by reset.
1 = Data read from internal registers, ROM, and RAM is driven onto the external multiplexed address/data bus.
0 = Data is not driven onto external bus.


BEGIN
RESET ?
YES
FETCH VECTOR FROM $XXFE & SXXFF
NO
ILLEGAL OPCODE ?
YES
FETCH VECTOR FROM $XXF8 & $XXF9
NO
CC REG I-BIT SET ?
YES
SWI ?
YES
FETCH VECTOR FROM $XXF6 & $XXF7
NO
NO VALID INTERRUPT REQUEST
NO
TIMER INTERRUPT ?
YES
FETCH VECTOR FROM $XXF2 & $XXF3
NO
SCSI SEL PHASE INTERRUPT ?
YES
FETCH VECTOR FROM $XXF0 & $XXF1
NO
A
A
SCSI ARBITRATION BEGUN INTERRUPT ?
YES
FETCH VECTOR FROM $XXEE & $XXEF
NO
SCSI ATTENTION INTERRUPT ?
YES
FETCH VECTOR FROM $XXEC & $XXED
NO
BUFFER ERROR INTERRUPT ?
YES
FETCH VECTOR FROM $XXEA & $XXEB
NO
BUFFER DONE INTERRUPT ?
YES
FETCH VECTOR FROM $XXE8 & $XXE9
NO
SERIAL INTERRUPT ?
YES
FETCH VECTOR FROM $XXE6 & $XXE7
NO
SWI ?
YES
FETCH VECTOR FROM $XXF6 & $XXF7
NO
NO VALID INTERRUPT REQUEST

NOTE:
XX is FF if mode is single-chip or expanded 1;
XX is BF if mode is expanded 2 or expanded 3.

Figure 8. Interrupt Handler Flowchart

MODE — Read-Only Status Bit Indicating Current Operating Mode

1 = Expanded 2 or expanded 3 mode
0 = Single-chip or expanded 1 mode

PSEL3–0 — Priority Select

These bits contain a binary number indicating the I-interrupt source with the highest priority. This binary code and its meaning are shown in Table 6.

Table 6. Interrupt Source and Priority

0110 — Timer	1010 — Buffer Error
0111 — SCSI Selection	1011 — Buffer Done
1000 — SCSI Arbitration	1100 — Serial Unit
1001 — SCSI Attention	

All other codes are reserved.

LOW POWER MODE: WAIT

For low power operation, this HDC implements the WAIT instruction. WAIT causes the CPU core of the HDC to assume a low power state, or wait mode, with the internal clocks still active.

In the wait mode, the machine state is stacked and program execution stops. The serial data controller, ECC unit, data buffer unit, host interface and timer are not affected. $\overline{\text{RESET}}$ and the I interrupt cause an exit from the wait mode if the I bit in the condition code register is clear.

TIMER

Preceded by a fixed divide-by-four prescaler, the programmable timer can be used for many purposes since its pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 9.

Because the timer has a 16-bit architecture, each specific functional segment, or capability, is represented by two 8-bit registers. These registers contain the high or low byte of that functional segment and are called the high-byte register and low-byte register, respectively. Generally, accessing the low-byte register of a specific timer function allows full control of that function. However, an access of the high-byte register inhibits that specific timer function until the low-byte register is also accessed.

NOTE

> The I bit in the condition code register should be set while manipulating both the high-byte and the low-byte registers of a specific timer function to ensure that an interrupt does not occur. Setting this bit prevents interrupts from occurring between the time that the high- and low-byte registers are accessed.

TIMER CONTROL REGISTER (TCR) $0012

The TCR is an 8-bit read/write register which contains five control bits. Three of these bits, ICIE, OCIE, and TOIE, control the interrupts associated with each of the three flag bits found in the timer status register (TSR): ICF, OCF, and TOF, respectively. The IEDG bit controls whether the negative or positive edge of the signal is significant to the input capture edge detector. The OLVL bit controls the next value to be clocked to the output level register in response to a successful output compare.

The TCR and the free-running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and remains low until raised high by a valid compare. The TCR is shown below:

	7	6	5	4	3	2	1	0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET	0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable Bit

1 = The timer interrupt is enabled when the input capture flag (ICF) status flag is set.
0 = Interrupt is inhibited.

OCIE — Output Compare Interrupt Enable Bit

1 = The timer interrupt is enabled when the output compare flag (OCF) status flag is set.
0 = Interrupt is inhibited.

TOIE — Timer Overflow Interrupt Enable Bit

1 = The timer interrupt is enabled when the timer overflow flag (TOF) status flag is set.
0 = Interrupt is inhibited.

Bits 4–2 — Not Used.

IEDG — Input Edge Bit

The input edge bit determines which level transition will trigger a free-running counter transfer to the input capture register.

1 = Positive edge
0 = Negative edge

OLVL — Output Level Bit

The output level bit is clocked into the output level register by the next successful output compare. This bit and the output level register are cleared by reset.

1 = High output
0 = Low output

TIMER STATUS REGISTER (TSR) $0013

The TSR is an 8-bit register of which the three most significant bits contain read-only status information. These three bits indicate the following:

- — A proper transition has taken place with an accompanying transfer of the free-running counter contents to the input capture register.
- — A match has been found between the free-runnign counter and the output compare register.
- — A free-running counter transition from $FFFF to $0000 has been sensed (timer overflow).

	7	6	5	4	3	2	1	0
	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

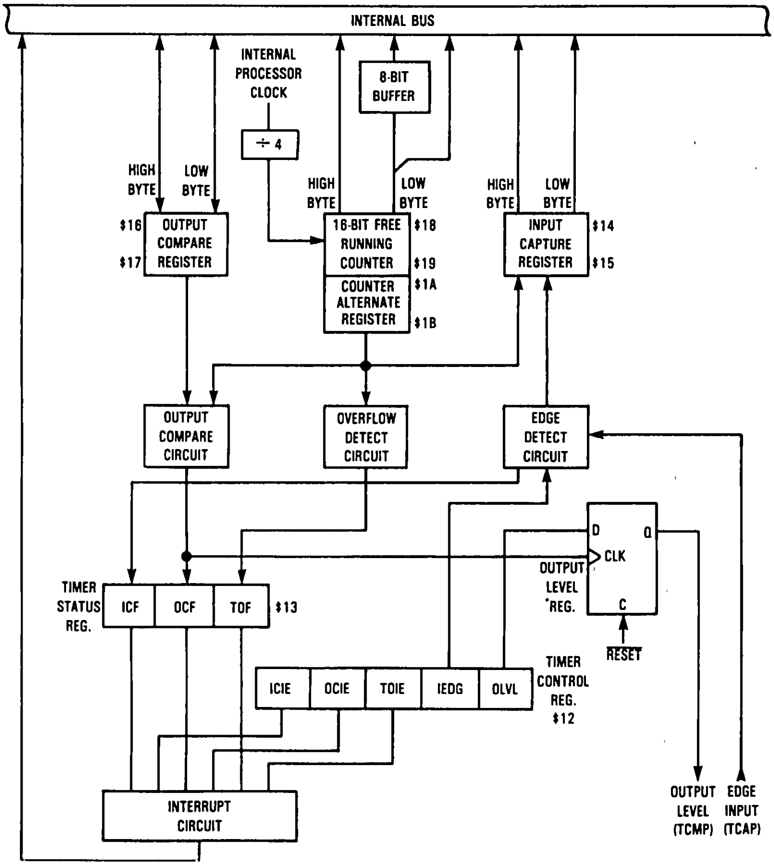

INTERNAL BUS
INTERNAL PROCESSOR CLOCK
÷ 4
8-BIT BUFFER
HIGH BYTE
LOW BYTE
$16
$17
OUTPUT COMPARE REGISTER
HIGH BYTE
LOW BYTE
16-BIT FREE RUNNING COUNTER
$18
$19
COUNTER ALTERNATE REGISTER
$1A
$1B
HIGH BYTE
LOW BYTE
INPUT CAPTURE REGISTER
$14
$15
OUTPUT COMPARE CIRCUIT
OVERFLOW DETECT CIRCUIT
EDGE DETECT CIRCUIT
D
Q
CLK
C
OUTPUT LEVEL REG.
RESET
TIMER STATUS REG.
ICF
OCF
TOF
$13
ICIE
OCIE
TOIE
IEDG
OLVL
TIMER CONTROL REG. $12
INTERRUPT CIRCUIT
OUTPUT LEVEL (TCMP)
EDGE INPUT (TCAP)

Figure 9. Timer Block Diagram

ICF — Input Capture Flag
- 1 = A proper edge has been sensed by the input capture edge detector.
- 0 = The flag is cleared by a processor access of the TSR (with the ICF set) followed by accessing the low byte ($15) of the input capture register.

OCF — Output Compare Flag
- 1 = The output compare register contents match the contents of the free-running counter.
- 0 = The flag is cleared by accessing the timer status register (with OCF set) and then accessing the low byte ($17) of the output compare register.

TOF — Timer Overflow Flag
- 1 = A transition of the free-running counter from $FFFF to $0000 occurred.
- 0 = The flag is cleared by accessing the timer status register (with TOF set) and then accessing the low byte of the free-running counter $19).

Bits 4–0 — Not used.

Accessing the TSR satisfies the first step required to clear any status bits that happen to be set during the access. The next step is to access the register associated with the status bit. Typically, this procedure is accomplished using the input capture and output compare functions.

A problem can occur when the user is using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the TOF could unintentionally be cleared if:

1) The TSR is read or written when TOF is set and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

Since the counter alternate register contains the same value as the counter, this alternate register can be read at any time without affecting the TOF.

INPUT CAPTURE REGISTER (ICR)

The two 8-bit registers, which comprise the 16-bit ICR, are read-only registers. After the corresponding input-capture edge detector senses a defined transition, these registers are used to latch the value of the free-running counter. The level transition that triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the ICR.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal processor clock preceding the external transition. This delay is required for internal synchronization. Resolution is affected by the prescaler allowing the timer to increment only every four internal processor clock cycles.

The free-running counter contents are transferred to the ICR on each proper signal transition regardless of whether the ICF is set or clear. The ICR always contains the free-running counter value that corresponds to the most recent input capture.

After reading the ICRs MSB ($14), counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the input-capture software routine and its interaction with the main program to determine the minimum pulse period attainable. The free-running counter increments every four internal processor clock cycles due to the prescaler.

A read of the LSB ($15) of the ICR does not inhibit the running-counter transfer since the read and the transfer occur on opposite edges of the internal processor clock. Minimum pulse periods are ones which allow software to read the LSB ($15) and perform needed operations.

OUTPUT COMPARE REGISTER (OCR)

The OCR is a 16-bit register, which is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The OCR can be used for such pruposes as indicating when a period of time has elapsed. The OCR is unique in that all bits are readable and writable and are not altered by the timer hardware. Reset does not affect the contents of this register. If the compare function is not needed, the two bytes of the OCR can be used as storage locations.

OCR contents are compared with the contents of the free-running counter every four internal processor clocks. If a match is found, the corresponding OCF bit of the TCR is set, and the corresponding OLVL bit is clocked to an output level register by the output compare circuit pulse. The values in the OCR and the OLVL should be changed after each successful comparison in order to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the OCR containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal processor clock cycles due to the internal prescaler. The minimum time required to update the OCR is a function of the program rather than the internal hardware.

A processor write may be made to either byte of the OCR without affecting the other byte. The OLVL is clocked to the output level register regardless of whether the OCF is set or clear.

Because neither the OCF nor OCR is affected by reset, care must be exercised when initializing the output compare function. The following procedure is recommended:

1) Write to the high byte of the OCR to inhibit further compares until the low byte is written,
2) Read the timer status register to arm the OCF if it is already set, and
3) Write to the low byte of the OCR to enable the output compare function with the flag clear.

This procedure is advantageous because it prevents the OCF bit from being set between the time OCF is read and OCR is written. A software example is as follows:

```
B7   16   STA  OCMPHI     Inhibit Output Compare
B6   13   LDA  TSTAT      Arm OCF Bit If Set
BF   17   STX  OCMPLD     Ready for Next Compare
```

COUNTER

The key element in the programmable timer is a 16-bit free-running counter, or counter register, preceded by a prescaler which divides the internal procesor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal processor clock is 2.0 megahertz. The counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting the value.

The double-byte free-running counter can be read from either the two locations $18–$19 (called counter register at this location), or $1A–$1B (counter alternate register at this location). A read from only the LSB of the free-running counter ($19, $1B) will receive the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the MSB ($18, $1A), then the read causes the LSB ($19, $1B) to be transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B). This access completes a read sequence of the total counter value. If either the free-running counter or counter alternate register MSB is read, the LSB must also be read in order to complete the sequence.

The free-running counter is configured to $FFF9 during reset and is always a read-only register. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 processor internal clock cycles. When the counter rolls over from $FFFF to $0000, TOF is set. An interrupt can also be enabled when counter rollover occurs by setting TOIE.

INSTRUCTION SET

The CPU of the MC68HC99 HDC is an MC68HC11 microprocessor. In addition to its ability to execute all MC6800 and MC6801 instructions, the HDC has an opcode map with a total of 91 new opcodes. Major func-

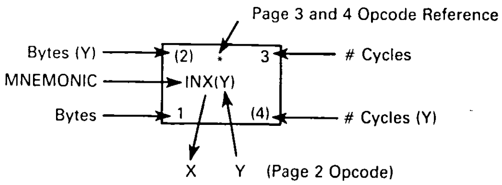
Table 4. Opcode Map

									ACCA				ACCB				
	INH		REL	INH	ACCA	ACCB	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	IMM	DIR	(Y) INDX	EXT	
HI / LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LOW
0 0000	TEST 1	SBA 2 1	BRA 3 2	(2) TSX(Y) 3 1 (4)	NEGA 2 1	NEGB 2 1	(3) NEG 6 2 (7)	NEG 6 3	SUBA 2 2	SUBA 3 2	(3) SUBA 4 2 (5)	SUBA 4 3	SUBB 2 2	SUBB 3 2	(3) SUBB 4 2 (5)	SUBB 4 3	0
1 0001	NOP 2 1	CBA 2 1	BRN 3 2	INS 3 1					CMPA 2 2	CMPA 3 2	(3) CMPA 4 2 (5)	CMPA 4 3	CMPB 2 2	CMPB 3 2	(3) CMPB 4 2 (5)	CMPB 4 3	1
2 0010	IDIV 41 1	BRSET 6 4	BHI 3 2	PULA 4 1					SBCA 2 2	SBCA 3 2	(3) SBCA 4 2 (5)	SBCA 4 3	SBCB 2 2	SBCB 3 2	(3) SBCB 4 2 (5)	SBCB 4 3	2
3 0011	FDIV 41 1	BRCLR 6 4	BLS 3 2	PULB 4 1	COMA 2 1	COMB 2 1	(3) COM 6 2 (7)	COM 6 3	*SUBD 4 3	*SUBD 5 2	(3) *SUBD 6 2 (7)	*SUBD 6 3	ADDD 4 3	ADDD 5 2	(3) ADDD 6 2 (7)	ADDD 6 3	3
4 0100	LSRD 3 1	BSET 6 3	(BHS) BCC 3 2	DES 3 1	LSRA 2 1	LSRB 2 1	(3) LSR 6 2 (7)	LSR 6 3	ANDA 2 2	ANDA 3 2	(3) ANDA 4 2 (5)	ANDA 4 3	ANDB 2 2	ANDB 3 2	(3) ANDB 4 2 (5)	ANDB 4 3	4
5 0101	(LSLD) ASLD 3 1	BCLR 6 3	(BLO) BCS 3 2	(2) TX(Y)S 3 1 (4)					BITA 2 2	BITA 3 2	(3) BITA 4 2 (5)	BITA 4 3	BITB 2 2	BITB 3 2	(3) BITB 4 2 (5)	BITB 4 3	5
6 0110	TAP 2 1	TAB 2 1	BNE 3 2	PSHA 3 1	RORA 2 1	RORB 2 1	(3) ROR 6 2 (7)	ROR 6 3	LDAA 2 2	LDAA 3 2	(3) LDAA 4 2 (5)	LDAA 4 3	LDBB 2 2	LDBB 3 2	(3) LDBB 4 2 (5)	LDBB 4 3	6
7 0111	TPA 2 1	TBA 2 1	BEQ 3 2	PSHB 3 1	ASRA 2 1	ASRB 2 1	(3) ASR 6 2 (7)	ASR 6 3		STAA 3 2	(3) STAA 4 2 (5)	STAA 4 3		STBB 3 2	(3) STBB 4 2 (5)	STBB 4 3	7
8 1000	(2) INX(Y) 3 1 (4)	PAGE 2	BVC 3 2	(2) PULX(Y) 5 1 (6)	ASLA 2 1	ASLB 2 1	(3) ASL 6 2 (7)	ASL 6 3	EORA 2 2	EORA 3 2	(3) EORA 4 2 (5)	EORA 4 3	EORB 2 2	EORB 3 2	(3) EORB 4 2 (5)	EORB 4 3	8
9 1001	(2) DEX(Y) 3 1 (4)	DAA 2 1	BVS 3 2	RTS 5 1	ROLA 2 1	ROLB 2 1	(3) ROL 6 2 (7)	ROL 6 3	ADCA 2 2	ADCA 3 2	(3) ADCA 4 2 (5)	ADCA 4 3	ADCB 2 2	ADCB 3 2	(3) ADCB 4 2 (5)	ADCB 4 3	9
A 1010	CLV 2 1	PAGE 3	BPL 3 2	(2) ABX(Y) 3 1 (4)	DECA 2 1	DECB 2 1	(3) DEC 6 2 (7)	DEC 6 3	ORAA 2 2	ORAA 3 2	(3) ORAA 4 2 (5)	ORAA 4 3	ORAB 2 2	ORAB 3 2	(3) ORAB 4 2 (5)	ORAB 4 3	A
B 1011	SEV 2 1	ABA 2 1	BMI 3 2	RTI 12 1					ADDA 2 2	ADDA 3 2	(3) ADDA 4 2 (5)	ADDA 4 3	ADDB 2 2	ADDB 3 2	(3) ADDB 4 2 (5)	ADDB 4 3	B
C 1100	CLC 2 1	(4) BSET 7 3 (8)	BGE 3 2	(2) PSHX(Y) 4 1 (5)	INCA 2 1	INCB 2 1	(3) INC 6 2 (7)	INC 6 3	(4) CPX(Y) 4 3 5	(3) CPX(Y) 5 2 (6)	(3) *CPX(Y) 6 2 (7)	(4) CPX(Y) 6 3 (7)	LDD 3 3	LDD 4 2	(3) LDD 5 2 (6)	LDD 5 3	C
D 1101	SEC 2 1	(4) BCLR 7 3 (8)	BLT 3 2	MUL 10 1	TSTA 2 1	TSTB 2 1	(3) TST 6 2 (7)	TST 6 3	BSR 6 2	JSR 5 2	(3) JSR 6 2 (7)	JSR 6 3	PAGE 4	STD 4 2	(3) STD 5 2 (6)	STD 5 3	D
E 1110	CLI 2 1	(5) BRSET 7 4 (8)	BGT 3 2	WAI 12 1			(3) JMP 3 2 (4)	JMP 3 3	LDS 3 3	LDS 4 2	(3) LDS 5 2 (6)	LDS 5 3	(4) LDX(Y) 3 3 (4)	(3) LDX(Y) 4 2 (5)	(3) LDX(Y) 5 2 (6)	(4) *LDX(Y) 5 3 (6)	E
F 1111	SEI 2 1	(5) BRCLR 7 4 (8)	BLE 3 2	SWI 14 1	CLRA 2 1	CLRB 2 1	(3) CLR 6 2 (7)	CLR 6 3	(2) XGDX(Y) 3 1 (4)	STS 4 2	(3) STS 5 2 (6)	STS 5 3		(3) STX(Y) 4 2 (5)	(3) STX(Y) 5 2 (6)	(4) *STX(Y) 5 3 (6)	F

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

***Page 3 and 4 Opcode Reference**

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



tional additions include a second 16-bit index register, two types of 16-by-16 divide instructions, bit manipulation instructions, and a WAIT instruction.

ADDRESSING MODES

Six addressing modes can be used to reference memory. Some instructions require an additional byte before the opcode to accomodate a multipage opcode map. This byte is called a prebyte. The following paragraphs provide a description of each addressing mode plus a discussion of the prebyte. In these descriptions, the term "effective address" is used to indicate the address in memory from which the argument is fetched or stored or from which execution proceeds.

IMMEDIATE ADDRESSING

In the immediate addressing mode, the actual argument is contained in byte(s) immediately following the instruction, where the number of bytes matches the size of the register. If prebyte is required, then these are two-, three-, or four-byte instructions.

DIRECT ADDRESSING

In the direct addressing mode, the LSB of operand address is contained in a single byte following the opcode, and the MSB is assumed to be $00. Direct addressing allows the user to access $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the HDC, the first 192 bytes of the internal RAM and registers are fixed in page zero of memory, with the exception of the bytes at $0000 through $0003 for the parallel ports. These bytes can be either internal or external depending on the operating mode.

EXTENDED ADDRESSING

In the extended addressing mode, the second and third bytes following the opcode contain the absolute address of the operand. If prebyte is required, then these are three- or four-byte instructions. One or two bytes are required for the opcode and two bytes for the effective address.

INDEXED ADDRESSING

In the indexed addressing mode, one index register (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors:

1) Current contents of the index register (X or Y) being used and
2) The 8-bit unsigned offset contained in the instruction.

This addressing mode allows referencing any memory location in the 64K byte address space. If prebyte is required, then these are usually two- or three-byte instructions, to accomodate the opcode plus an 8-bit offset.

INHERENT ADDRESSING

In the inherent addressing mode, all information to execute an instruction is contained in the opcode. Operands, if any, are registers, and no memory reference is required. These are usually one- or two-byte instructions.

RELATIVE ADDRESSING

The relative addressing mode is used for branch instructions. If the branch condition is true, the contents of the eight-bit signed byte following the opcode, the offset, is added to the contents of the PC to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

PREBYTE

In order to expand the number of instructions available in the HDC, a prebyte instruction is used with certain instructions. The instructions affected are usually associated with the index register Y. Opcode instructions which do not require a prebyte could be considered as page 1 of the overall opcode map. The remaining opcodes could be considered as pages 2, 3, and 4 of the opcode map and would require a prebyte: $18 for page 2, $1A for page 3, and $CD for page 4.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to V_{DD} +0.5	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain per pin* Excluding V_{DD}, V_{SS}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leqslant (V_{in}$ or $V_{out}) \leqslant V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance PLCC	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{I/O}$

P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 5%; V_{SS} = 0 Vdc; T_A = 0°C to 70°C)

Characteristic	Symbol	Min	Max	Unit
Supply Voltage Range	V_{DD}	4.75	5.25	V
Output High Voltage I_{Load} = 0.8 mA	V_{OH}	2.4	—	V
Output Low Voltage SELOUT, BSYOUT, I/O/OUT, CMD, MSG, ARB/SDBEN, REQ, AMEN, WRCLK, NRZ I_{Load} = 1.6 mA	V_{OL}	—	0.4	V
All Other Outputs I_{Load} = 3.7 mA	V_{OL}	—	0.4	V
Input High Voltage	V_{IH}	—	2.0	V
Input Low Voltage	V_{IL}	0.8	—	V
Three-State Leakage $V_{in} = V_{IH}$ or V_{IL} (0.0 − 5.25 V)	I_{OZ}	—	±10	µA
Input Leakage Current	I_{IN}	—	±1	µA
Input Capacitance	C_{IN}	—	15	pF
Power Dissipation	P_D	—	TBD	mW
Total Current Supply				
CPU Only	C_{IDD}	TBD	TBD	V
CPU In Wait Mode	W_{IDD}	TBD	TBD	V
Everything Running	R_{IDD}	TBD	TBD	V

NOTES:

1. No DC loads, V_{IL} = 0.0 V, V_{IH} = 5.0 V, CLK = 10 MHz
2. REFCLK = 0.0 V, serial port and buffer are not active, buffer is refreshing DRAMs.
3. REFCLK = 25 MHz

EXPANSION BUS TIMING (see Figure 10)

Num	Characteristic	Formula		Symbol	Min	Max	Unit
	Frequency of Operation (E Clock Frequency)			f_o	2.5	2.5	MHz
1	Cycle Time			t_{cyc}	400	—	ns
2	Pulse Width, E Low	$1/2\ t_{cyc} - 32$		PW_{EL}	168	—	ns
3	Pulse Width, E High	$1/2\ t_{cyc} - 24$		PW_{EH}	176	—	ns
4	E and AS Rise and Fall Time			t_r, t_f	—	15	ns
9	Address Hold Time	$1/8\ t_{cyc} - 25$	(a)	t_{AH}	25	—	ns
12	Non-Muxed Address Valid Time to E Rise	$PW_{EL}\text{-}(t_{ASD} + 78)$	(b)	t_{AV}	60	—	ns
17	Read Data Setup Time			t_{DSR}	50	—	ns
18	Read Data Hold Time	$Max = t_{MAD}$		t_{DHR}	10	60	ns
19	Write Data Delay Time	$1/8\ t_{cyc} + 70$	(a)	t_{DDW}	—	120	ns
21	Write Data Hold Time	$1/8\ t_{cyc} - 30$	(a)	t_{DHW}	20	—	ns
22	Muxed Address Valid Time to E Rise	$PW_{EL} - (t_{ASD} + 78)$	(b)	t_{AVM}	60	—	ns
24	Muxed Address Valid Time to AS Fall	$1/8\ t_{cyc} - 35$	(a)	t_{ASL}	15	—	ns
25	Muxed Address Hold Time	$1/8\ t_{cyc} - 25$	(b)	t_{HL}	25	—	ns
26	Delay Time, E to AS Rise	$1/8\ t_{cyc} - 20$	(a)	t_{ASD}	30	—	ns
27	Pulse Width, AS High	$1/4\ t_{cyc} - 25$		PW_{ASH}	75	—	ns
28	Delay Time, AS to E Rise	$1/8\ t_{cyc} - 20$	(b)	t_{ASED}	30	—	ns
29	MPU Address Access Time	$t_{AVM} + t_r + PW_{EH} - t_{DSR}$	(b)	t_{ACCA}	201	—	ns
35	MPU Access Time	$PW_{EH}\text{-}t_{DSR}$		t_{ACCE}	—	126	ns
36	Muxed Address Delay (Previous Cycle MPU Read)	$t_{ASD} + 30$	(a)	t_{MAD}	60	—	ns

NOTE:

Input clocks with duty cycles other than 50% affect bus performance. Timing parameters affected by the input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions for 1/8 t_{cyc} in the formulas the expansion bus timing table where applicable.

(a) $(1\text{-}DC) \times 1/4\ t_{cyc}$

(b) $DC \times 1/4\ t_{cyc}$

where:

DC is the decimal value of the duty cycle percentage, (high time)

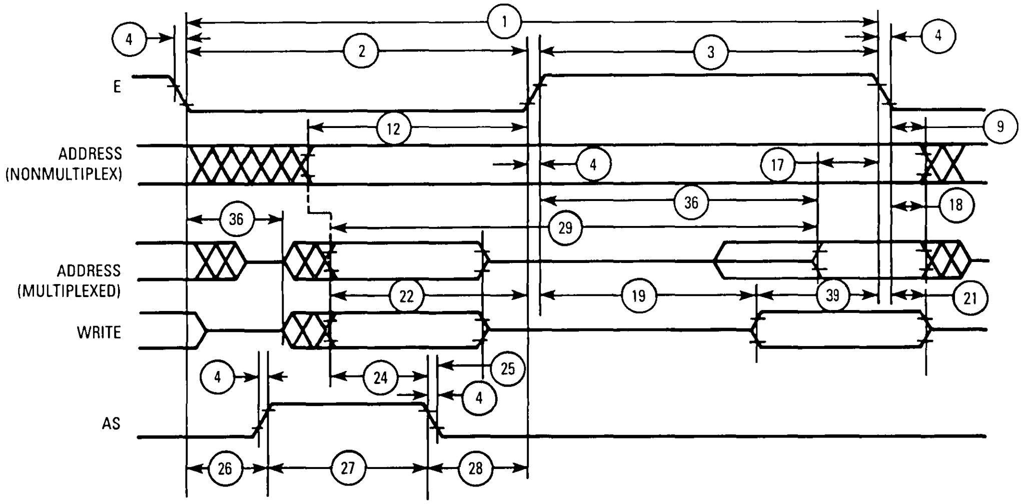
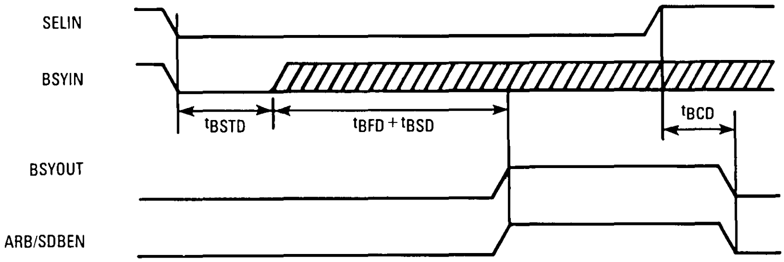


Figure 10. Expansion Bus Timing Diagram

SCSI SELECTION AND RESELECTION TIMING (see Figures 11 and 12)

Characteristic	Symbol	Min	Max	Unit
Selection Phase Detect Time	t_{SPD}	1	1.25	t_{cyc}
Bus Settle Delay (BSYIN and SELIN Negated)	t_{BSTD}	t_{cyc}	—	ns
Bus Free Delay	t_{BFD}	2 t_{cyc}	—	ns
Bus Set Delay	t_{BSD}	1/2	2.0	t_{cyc}
Bus Clear Delay (ARB/SDBEN, BSYOUT Negated)	t_{BCD}	0	1.5 t_{cyc}	ns



NOTE:
The assertion of SELIN during the arbitration phase will asynchronously terminate the arbitration phase and cause the HDC to negate BSYOUT and ARB/SDBEN. The assertion of SELOUT terminates the arbitration phase.

Figure 11. SCSI Arbitration Timing Diagram

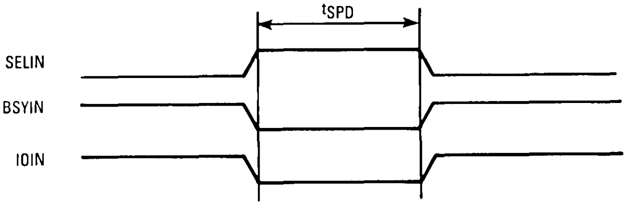


Figure 12. HDC SCSI Selection Phase Timing Diagram

HOST INTERFACE TRANSFER TIMING (see Figures 13 and 14)

Characteristic	Symbol	Min	Max	Unit
Request asserted to ACK Asserted	t_{RAAA}	0	—	ns
ACK Asserted to REQ Negated (Note 1)	t_{AARN}	0	1/8 t_{cyc}	ns
REQ Negated to ACK Negated	t_{RNAN}	0	—	ns
ACK Negated to REQ Asserted (Note 2)	t_{ANRA}	1/2	1.5	t_{cyc}
Data Valid to REQ Asserted (Transmit)	t_{DVR}	55	—	ns
Data Hold from ACK Asserted (Transmit)	t_{DHAA}	0	—	ns
Data Valid to ACK Asserted (Receive)	t_{DVAA}	55	—	ns
REQ Negated to Data Invalid (Hold Time, Receive)	t_{RNDI}	0	—	ns

NOTES:
1. See note (a) in **EXPANSION BUS TIMING** table.
2. Both edges of ACK can affect the host transfer speed. Slow ACK timing causes the transfer speed to be degraded in integer multiples of t_{cyc}.

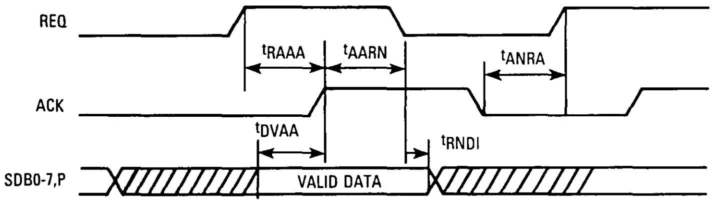

REQ
ACK
SDB0-7,P
tRAAA
tAARN
tANRA
tDVAA
tRNDI
VALID DATA

Figure 13. HDC SCSI Receive Timing Diagram

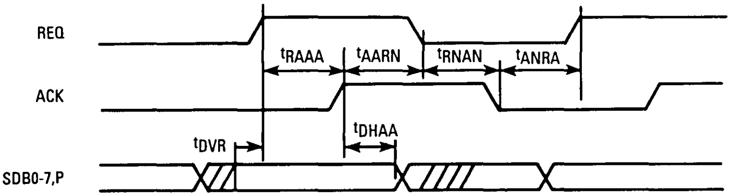

REQ
ACK
SDB0-7,P
tRAAA
tAARN
tRNAN
tANRA
tDVR
tDHAA

Figure 14. HDC SCSI Transmit Timing Diagram

DISK READ/WRITE TIMING (see Figures 15 and 16)

Parameter	Symbol	10 MHz		16 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
REFCLK Period	t_{RFCP}	100	4000	62	4000	40	4000	ns
REFCLK Pulse Width (50 ± 9.2% Duty Cycle)	PW_{RCL}	41	59	25	37	16	24	ns
REFCLK Rise/Fall Time	t_{RCRF}	—	10	—	10	—	10	ns
NRZ Read Data Setup Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{RDS}	20	—	13	—	13	—	ns
NRZ Read Data Hold Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{RDH}	20	—	13	—	13	—	ns
ST-506 AMF Setup Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{AMFS}	20	—	13	—	13	—	ns
WRCLK Pulse Width (50 ± 18.8% Duty Cycle)	PW_{WCL}	31	69	19	43	12	28	ns
NRZ Write Data Setup Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{WDS}	14	—	8	—	8	—	ns
NRZ Write Data Hold Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{WDH}	14	—	8	—	8	—	ns
ST-506 AMEN Setup Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{AMES}	14	—	8	—	8	—	ns
ST-506 AMEN Hold Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{AMEF}	14	—	8	—	8	—	ns

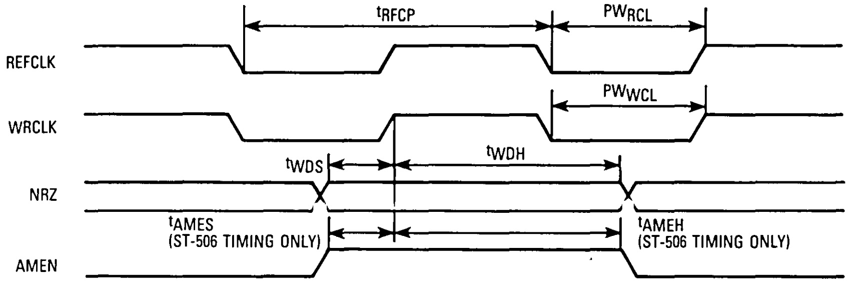

tRFCP
PWRCL
REFCLK
PWWCL
WRCLK
tWDS
tWDH
NRZ
tAMES
(ST-506 TIMING ONLY)
tAMEH
(ST-506 TIMING ONLY)
AMEN

Figure 15. Disk Read Timing Diagram

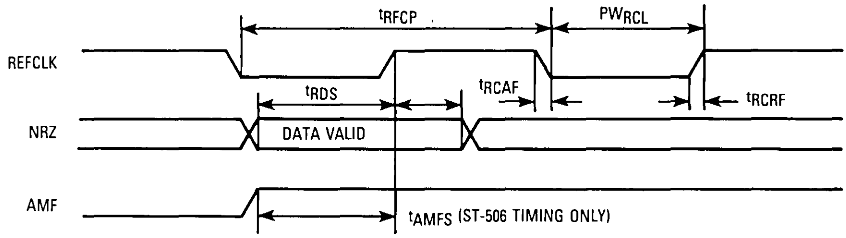

tRFCP
PWRCL
REFCLK
tRDS
tRCAF
tRCRF
NRZ
DATA VALID
AMF
tAMFS (ST-506 TIMING ONLY)

Figure 16. Disk Write Timing Diagram

SPECIAL TIMING (see Figures 17 and 18)

Characteristic	Symbol	Min	Max	Unit
Reset Low Input Pulse Width	PW_{RST}	4	—	t_{cyc}
Mode/Parity/Unit Programming Setup	t_{MPS}	2	—	t_{cyc}
Mode/Parity/Unit Programming Hold	t_{MPH}	0	1	t_{cyc}
Timer Input Capture PW (Single-Chip Mode)	PW_{TIM}	200	—	ns

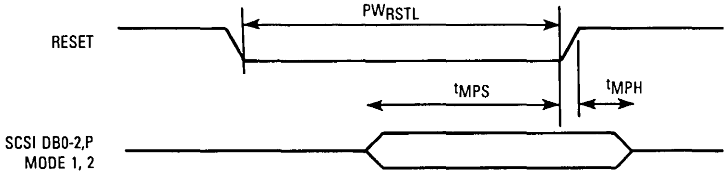

PWRSTL
RESET
tMPS
tMPH
SCSI DB0-2,P
MODE 1, 2

Figure 17. Reset Timing Diagram

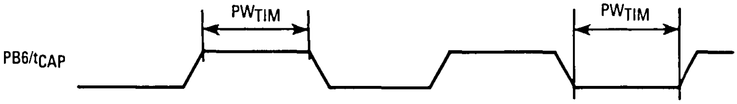

PWTIM
PWTIM
PB6/tCAP

Figure 18. Input Capture Timing Diagram
(Single-Chip Mode)

PORT READ AND WRITE TIMING (see Figures 19 and 20)

Parameter	Symbol	Min	Max	Unit
Delay Time, Peripheral Data Write	t_{PWD}	—	75	ns
Peripheral Read Data Setup Time				ns
PB0-PB7, PC0-PC7	t_{PDSU1}	50	—	
SDB0-SDB7, P	t_{PDSU2}	0	—	
SCSI Inputs	t_{PDSU3}	125	—	
Peripheral Read Data Hold Time				ns
PB0-PB7, PC0-PC7	t_{PDH1}	25	—	
SDB0-SDB7, P	t_{PDH2}	0	—	
SCSI Inputs	t_{PDH3}	125	—	

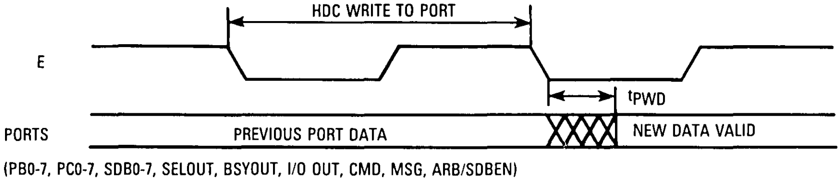

HDC WRITE TO PORT
E
tPWD
PORTS
PREVIOUS PORT DATA
NEW DATA VALID
(PB0-7, PC0-7, SDB0-7, SELOUT, BSYOUT, I/O OUT, CMD, MSG, ARB/SDBEN)

Figure 19. Port Write Timing Diagram

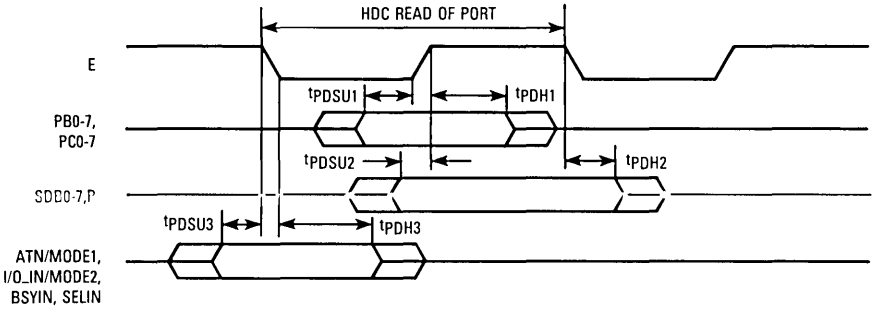

HDC READ OF PORT
E
tPDSU1
tPDH1
PB0-7,
PC0-7
tPDSU2
tPDH2
SDB0-7,P
tPDSU3
tPDH3
ATN/MODE1,
I/O_IN/MODE2,
BSYIN, SELIN

Figure 20. Port Read Timing Diagram

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola using the following media:

MDOS, disk file
MS-DOS disk file (360K)
EPROM(s) 2516, 2716, 2532, 2732

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MDOS™ or MS™-DOS disk file) may be submitted for pattern generation. Disk should be programmed with the customer's program, using positive logic sense for address and data. The diskette should be clearly labeled with the customer's name, date, project or product name, and the filename containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MDOS Disk File

MDOS is Motorola's Disk Operating System available on the EXORciser® development system. The disk media submitted must be a single-sided, single-density, 8-inch MDOS compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .LO output of the M68HC11 cross assembler should be furnished. In addition, the file must be produced using the ROLLOUT command, so that it contains the absolute image of the M68HC99 memory. It is necessary to include the entire memory image of both program and data space. All unused bytes, including those in the user space, must be set to logic zero.

MS-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5-1/4 inch compatible floppy diskette. The diskette must contain the object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by cross assemblers and linkers on IBM PC style machines.

To contain the entire MC68HC99 program, 12K bytes of EPROM are necessary. Six 2516 or 2716 type EPROMs or three 2532 or 2732 type EPROMs can be submitted for pattern generation. The EPROM is programmed with the customer's program using positive logic sense for address and data. Submissions on two EPROMs must be clearly marked. All unused bytes, including the user's space, must be set to zero.

If the MC68HC99 HDC ROM pattern is submitted on three 2532 or 2732 EPROMs, or on six 2516 or 2716 type EPROMs, memory map addressing is one-for-one. The data space ROM runs from EPROM address $018 to $05F, and program space ROM runs from EPROM address $960 to $FF7, with vectors from $FFC to $FFF.

For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.

Verification Media

All original pattern media, EPROMs or floppy disks, are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM Verification Units (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

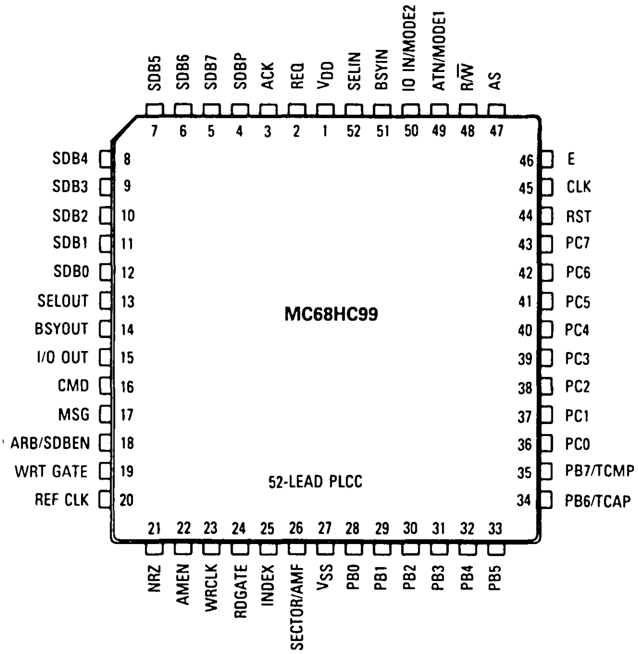
Ordering Information

The following table provides generic information pertaining to the package type, temperature, and order numbers for the MC68HC99.

Package Type	Temperature	Order Number
PLCC FN Suffix	0°C to 70°C −40°C to +85°C	MC68HC99FN MC68HC99CFN

MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
EXORciser is a registered trademark of Motorola Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENT


SDB5
SDB6
SDB7
SDBP
ACK
REQ
VDD
SELIN
BSYIN
IO IN/MODE2
ATN/MODE1
R/W̄
AS
7
6
5
4
3
2
1
52
51
50
49
48
47
SDB4 8
SDB3 9
SDB2 10
SDB1 11
SDB0 12
SELOUT 13
BSYOUT 14
I/O OUT 15
CMD 16
MSG 17
ARB/SDBEN 18
WRT GATE 19
REF CLK 20
46 E
45 CLK
44 RST
43 PC7
42 PC6
41 PC5
40 PC4
39 PC3
38 PC2
37 PC1
36 PC0
35 PB7/TCMP
34 PB6/TCAP
MC68HC99
52-LEAD PLCC
21
22
23
24
25
26
27
28
29
30
31
32
33
NRZ
AMEN
WRCLK
RDGATE
INDEX
SECTOR/AMF
VSS
PB0
PB1
PB2
PB3
PB4
PB5

Mechanical Data

Volume II

4

MECHANICAL DATA

This chapter contains the package availability, at the time of printing for each device as listed in Table 4-1 Package Cross-Reference List. Due to technology advances and customer requirements, case numbers may change from this listing. If more information is needed on packaging and dimensions, contact your local sales representative.

Table 4-1. Case Number Cross Reference Table

Device	Plastic (P)	Cerdip (S)	PLCC (FN)
MC2672	711-03	*	*
MC2674	711-03	*	*
MC6800	711-03	734-04	*
MC6801	711-03	734-04	*
MC6801U4	711-03	734-04	*
MC6802	711-03	734-04	*
MC6803	711-03	734-04	*
MC6803U4	711-03	734-04	*
MC6804J1	738-03	*	*
MC6804J2	738-03	*	*
MC6804P2	710-02	*	776-02
MC6805P2	710-02	733A-01	776-02
MC6805P6	710-02	733A-01	776-02
MC6805R2	711-03	734-04	777-02
MC6805R3	711-03	734-04	777-02
MC6805S2	710-02	*	*
MC6805S3	710-02	*	*
MC6805U2	711-03	734-04	777-02
MC6805U3	711-03	734-04	777-02
MC6809	711-03	734-04	*
MC6809E	711-03	734-04	*
MC6810	709-02	623-05	*
MC6821	711-03	734-04	*
MC6840	710-02	733-04	*
MC6844	711-03	734-04	*
MC6845	711-03	734-04	*
MC6850	709-02	623-05	*
MC6852	709-02	623-05	*
MC6854	710-02	733-04	*
MC6898	*	*	778-02
MC68488	711-03	734-04	*
MC68701	*	734A-01	*
MC68701U4	*	734A-01	*
MC68704P2	*	733A-01	*
MC68705P3	*	733A-01	*

*Not avılable in this package.

4

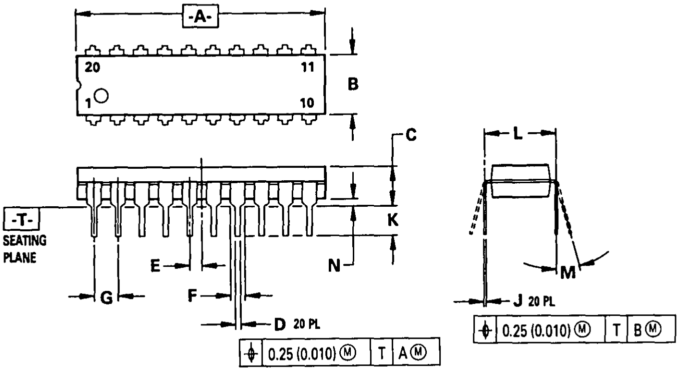
Table 4-1. Case Number Cross Reference Table (Continued)

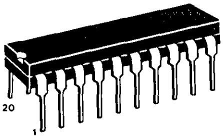
Device	Plastic (P)	Cerdip (S)	PLCC (FN)
MC68705P5	*	733A-01	*
MC68705R3	711-03	734-04	777-02
MC68705R5	711-03	734-04	777-02
MC68705S3	*	733-04	*
MC68705U3	*	734-04	*
MC68705U5	*	734-04	*
MC68HC04J2	738-03	*	*
MC68HC04J3	738-03	*	*
MC68HC04P3	710-02	*	*
MC68HC04P4	710-02	*	*
MC68HC05A6	711-03	*	777-02
MC68HC05B4	767-02	*	778-02
MC68HC05B6	767-02	*	778-02
MC68HC05C2	711-03	*	777-02
MC68HC05C3	711-03	*	777-02
MC68HC05C4	711-03	*	777-02
MC68HC05C8	711-03	*	777-02
MC68HC05L6	*	*	779-02
MC68HC05M4	*	*	778-02
MC68HC11A0	767-02	*	778-02
MC68HC11A1	767-02	*	778-02
MC68HC11A8	767-02	*	778-02
MC68HC11D3	767-02	*	778-02
MC68HC11E9	767-02	*	778-02
MC68HC24	711-03	*	*
MC68HC34	*	734-04	777A-01
MC68HC99	*	*	778-02
MC68HC704P4	*	733A-01	*
MC68HC705B5	767-02	740-03	778-02
MC68HC705C8	711-03	734-04	777-02
MC68HC805C4	711-03	*	777-02
MC68HC805B6	*	*	778-02
MC68HC811E2	711-03	*	778-02
MC68HCL05C4	711-03	*	777-02
MC68HCL05C8	711-03	*	777-02
MC68HSC05C4	711-03	*	777-02
MC68HSC05C8	711-03	*	777-02
MC146805E2	710-02	*	777-02
MC146805F2	710-02	733A-01	776-02
MC146805G2	711-03	*	777-02
MC146818	709-02	*	*
MC146818A	709-02	*	776-02
MC146823	711-03	*	777-02

*Not avilable in this package.

20-PIN PACKAGE

PLASTIC DIP
Case 738-03


-A-
20
11
1
10
B
C
-T-
SEATING PLANE
K
E
N
G
F
D 20 PL
0.25 (0.010) Ⓜ T AⓂ
L
M
J 20 PL
0.25 (0.010) Ⓜ T BⓂ



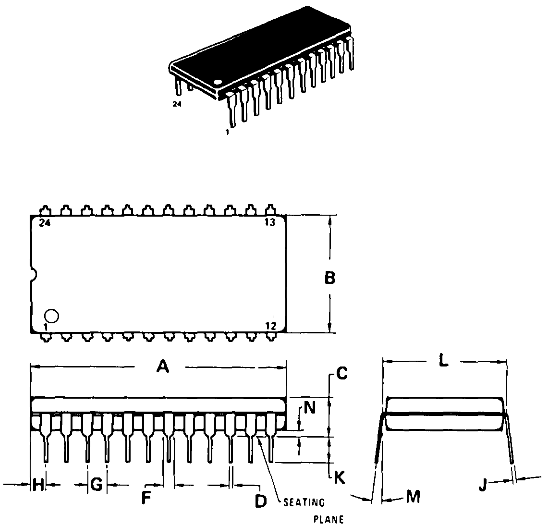
NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION "L" TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.66	27.17	1.010	1.070
B	6.10	6.60	0.240	0.260
C	3.81	4.57	0.150	0.180
D	0.39	0.55	0.015	0.022
E	1.27 BSC		0.050 BSC	
F	1.27	1.77	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.21	0.38	0.008	0.015
K	2.80	3.55	0.110	0.140
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.01	0.020	0.040

24-PIN PACKAGE

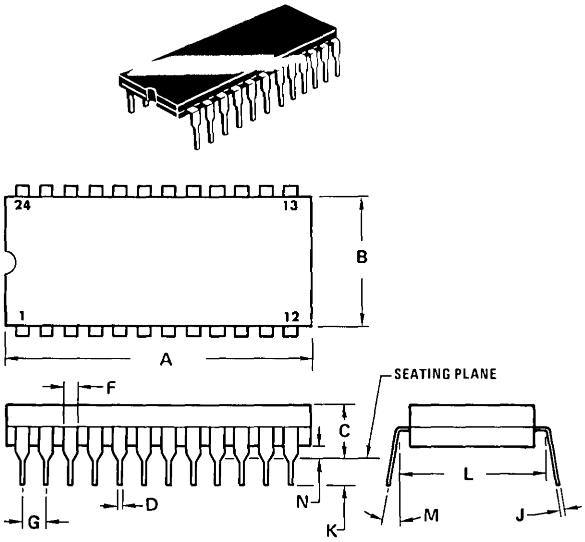
PLASTIC DIP
Case 709-02



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.37	32.13	1.235	1.265
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.03	0.065	0.080
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

CERAMIC PACKAGE
Case 623-05



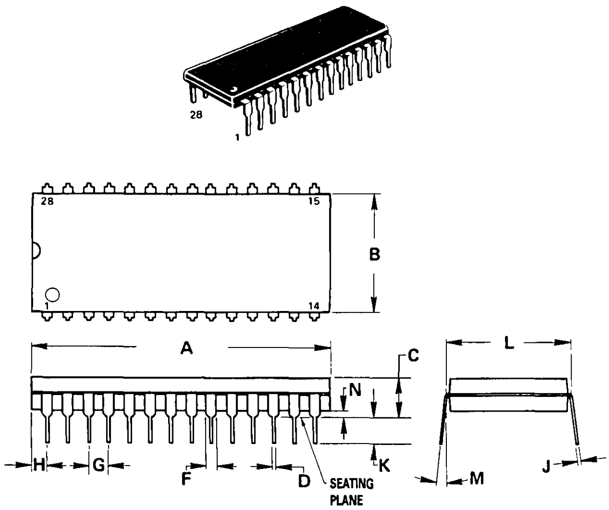
NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.77	1.230	1.290
B	12.70	15.49	0.500	0.610
C	4.06	5.59	0.160	0.220
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

28-PIN PACKAGE

PLASTIC PACKAGE

Case 710-02



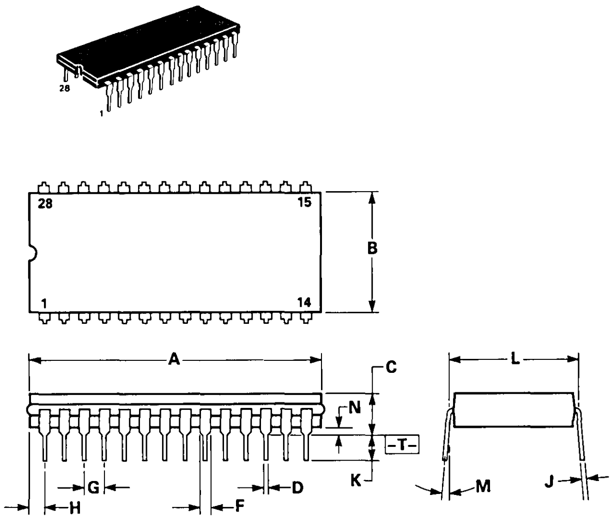
NOTES:

1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

CERAMIC PACKAGE

Case 733-04

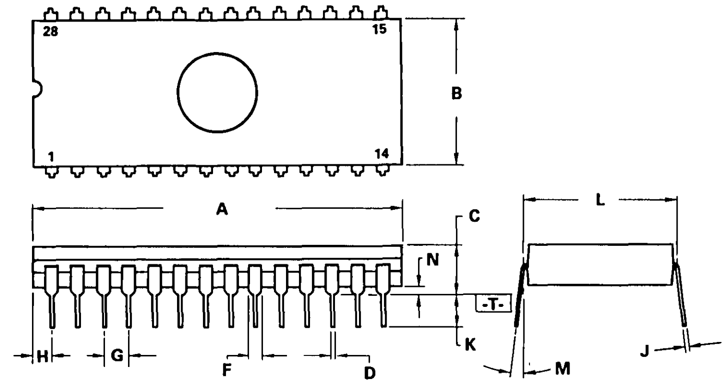


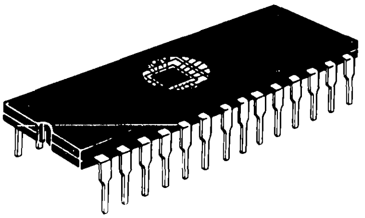
NOTES:

1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS:
   | ⌖ | ϕ 0.25 (0.010) Ⓜ | T | A Ⓜ |
3. -T- IS SEATING PLANE.
4. DIM A AND B INCLUDES MENISCUS.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING & TOLERANCING PER Y14.5, 1982.
7. CONTROLLING DIM: INCH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.84	1.435	1.490
B	12.70	15.36	0.500	0.605
C	4.06	5.84	0.160	0.230
D	0.38	0.55	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

CERAMIC PACKAGE
Case 733A-01





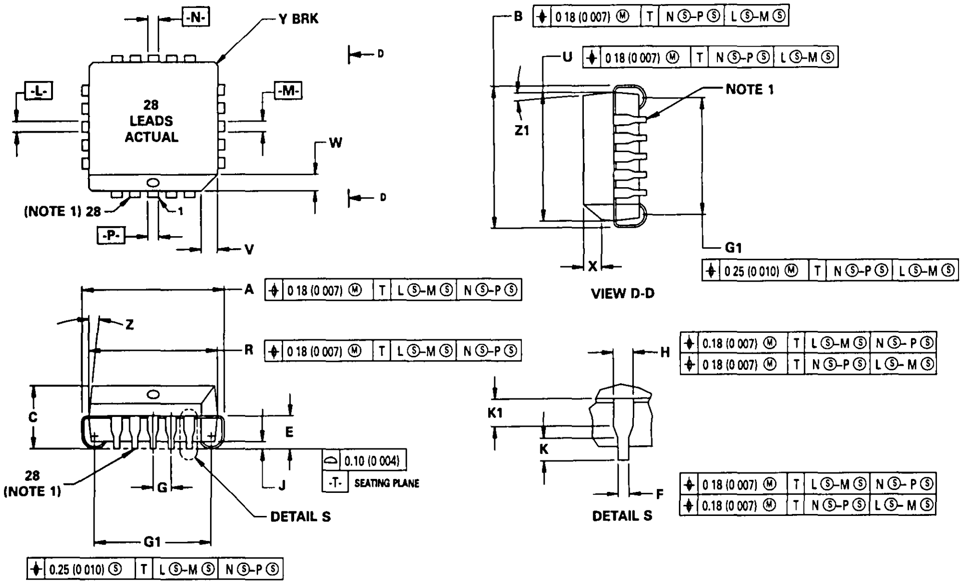
NOTES:

1. DIMENSION "A" IS A DATUM. T IS BOTH A DATUM AND A SEATING PLANE.
2. POSITIONAL TOLERANCE FOR LEADS: (28 PLACES)
   ⌖ | ⌀ 0.25 (0.010) Ⓜ | T | A Ⓜ
3. DIMENSIONS "A" & B INCLUDE MENISCUS.
4. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION· INCH.

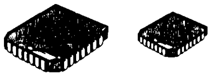
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.84	1.435	1.490
B	12.70	15 36	0.500	0.605
C	4.06	6 09	0.160	0.240
D	0.38	0.55	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.17	4.06	0.125	0.160
L	15.24 BSC		0 600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

PLCC PACKAGE

Case 776-02


-N-
Y BRK
D
-L-
-M-
28
LEADS
ACTUAL
W
D
(NOTE 1) 28
1
-P-
V
B ⌖ 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
U ⌖ 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
NOTE 1
Z1
G1
X
⌖ 0.25 (0.010) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
VIEW D-D
A ⌖ 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
Z
R ⌖ 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
C
E
0.10 (0.004)
-T- SEATING PLANE
J
28
(NOTE 1)
G
DETAIL S
G1
⌖ 0.25 (0.010) Ⓢ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
H
⌖ 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
⌖ 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
K1
K
F
⌖ 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
⌖ 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
DETAIL S

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	12.32	12.57	0.485	0.495
B	12.32	12.57	0.485	0.495
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	—	0.020	—
K	0.64	—	0.025	—
R	11.43	11.58	0.450	0.456
U	11.43	11.58	0.450	0.456
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	—	0.50	—	0.020
Z	2°	10°	2°	10°
G1	10.42	10.92	0.410	0.430
K1	1.02	—	0.040	—
Z1	2°	10°	2°	10°

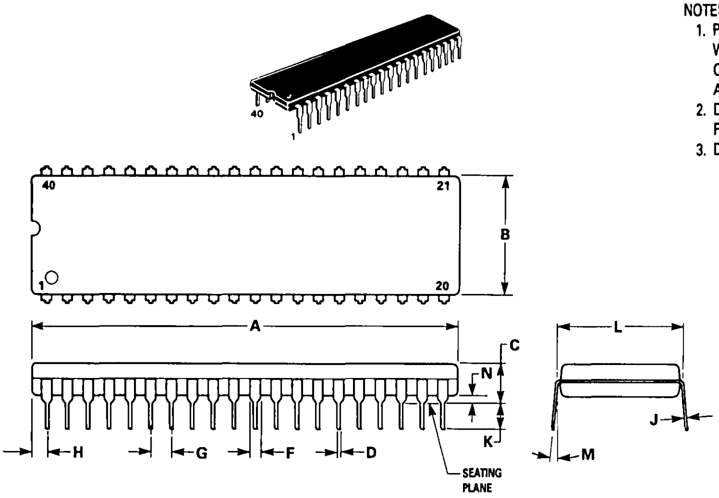


NOTES

1. **DUE TO SPACE LIMITATION, CASE 776-02 SHALL BE REPRESENTED BY A GENERAL (SMALLER) CASE OUTLINE DRAWING RATHER THAN SHOWING ALL 28 LEADS.**
2. DATUMS -L-, -M-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT PLASTIC BODY AT MOLD PARTING LINE
3. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE
4. DIM R AND U DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 (0.010) PER SIDE.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION INCH.

40-PIN PACKAGE

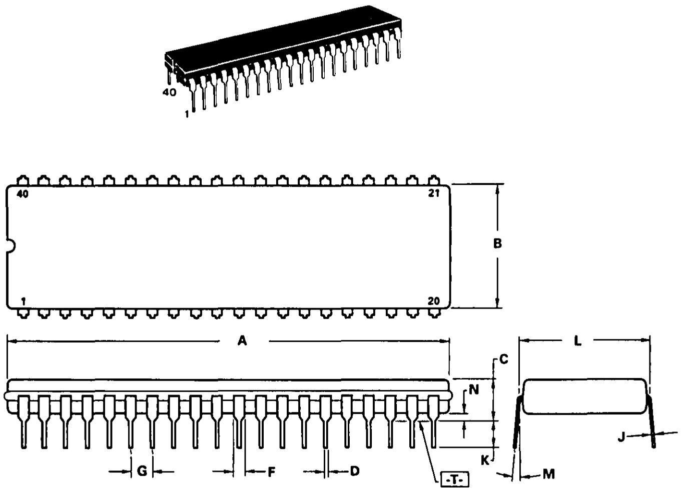
PLASTIC PACKAGE
Case 711-03



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2 065
B	13 72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0 022
F	1.02	1.52	0 040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0 600 BSC	
M	0°	15°	0°	15°
N	0 51	1.02	0 020	0 040

CERAMIC PACKAGE
Case 734-04

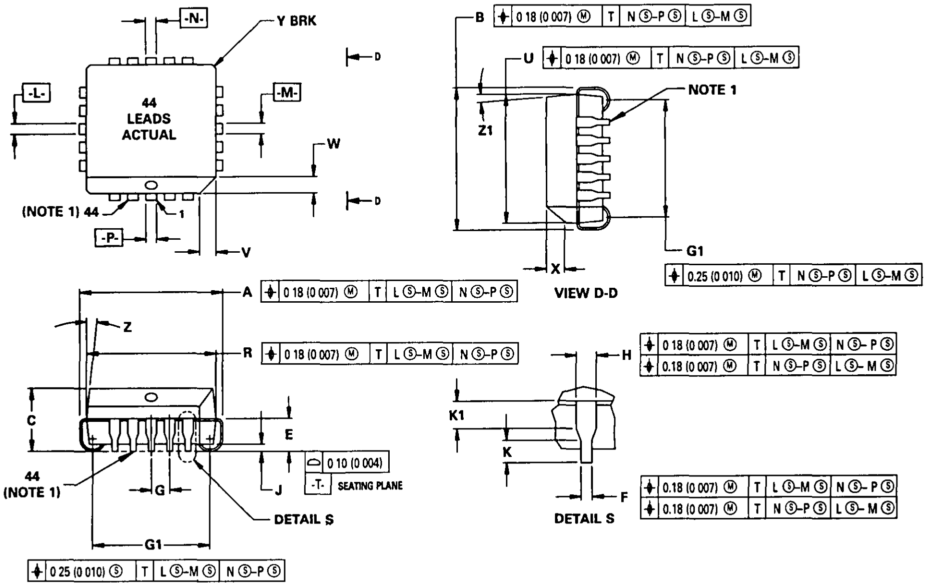


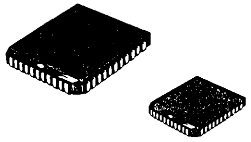
NOTES:
1. DIM -A- IS A DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:
   | ⌖ | ϕ 0.25 (0.010) Ⓜ | T | A Ⓜ |
3. -T- IS SEATING PLANE.
4. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONS A AND B INCLUDE MENISCUS.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	5.31	53.24	2.020	2.096
B	12.70	15.49	0.500	0.610
C	4.06	5 84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0 51	1.27	0.020	0.050

PLCC PACKAGE

Case 777-02


-N-
Y BRK
D
-L-
-M-
44
LEADS
ACTUAL
W
D
(NOTE 1) 44
1
-P-
V
B 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
U 0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
NOTE 1
Z1
G1
X
0.25 (0.010) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
VIEW D-D
A 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
Z
R 0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
C
E
0.10 (0.004)
-T- SEATING PLANE
J
44
(NOTE 1)
G
DETAIL S
G1
0.25 (0.010) Ⓢ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
H
0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
K1
K
F
0.18 (0.007) Ⓜ T L Ⓢ-M Ⓢ N Ⓢ-P Ⓢ
0.18 (0.007) Ⓜ T N Ⓢ-P Ⓢ L Ⓢ-M Ⓢ
DETAIL S



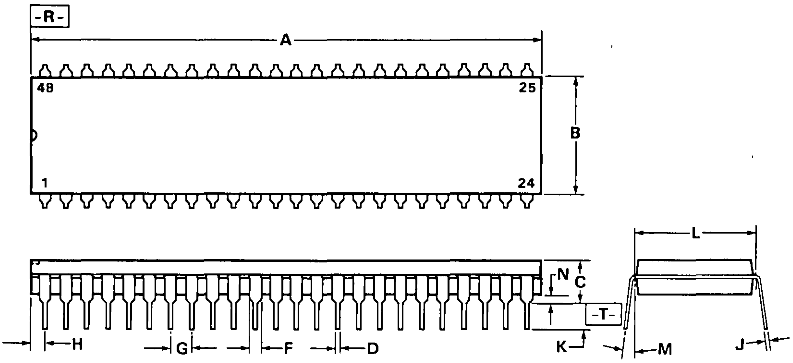
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	17.40	17.65	0.685	0.695
B	17.40	17.65	0.685	0.695
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	—	0.020	—
K	0.64	—	0.025	—
R	16.51	16.66	0.650	0.656
U	16.51	16.66	0.650	0.656
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	—	0.50	—	0.020
Z	2°	10°	2°	10°
G1	15.50	16.00	0.610	0.630
K1	1.02	—	0.040	—
Z1	2°	10°	2°	10°

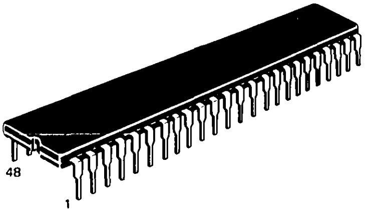
NOTES

1. **DUE TO SPACE LIMITATION, CASE 777-02 SHALL BE REPRESENTED BY A GENERAL (SMALLER) CASE OUTLINE DRAWING RATHER THAN SHOWING ALL 44 LEADS.**
2. DATUMS -L-, -M-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT PLASTIC BODY AT MOLD PARTING LINE.
3. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
4. DIM R AND U DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 (0.010) PER SIDE.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION: INCH.

48-PIN PACKAGE

PLASTIC PACKAGE
Case 767-02

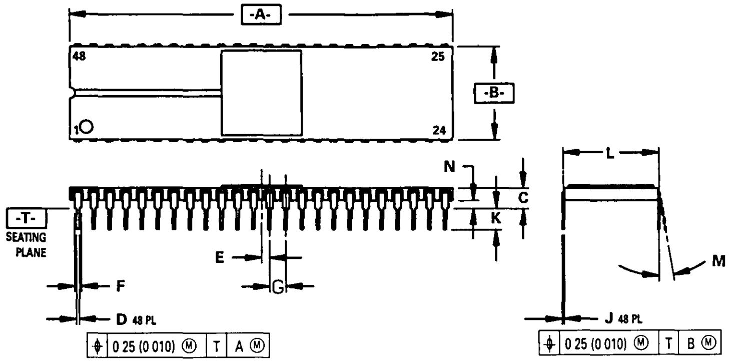




NOTES
1 -R- IS END OF PACKAGE DATUM PLANE
-T- IS BOTH A DATUM AND SEATING PLANE
2 POSITIONAL TOLERANCE FOR LEADS 1 AND 48
⌖ | 0 51 (0 020) | T | B Ⓜ | R
POSITIONAL TOLERANCE FOR LEAD PATTERN
⌖ | 0 25 (0 010) | T | B Ⓜ
3 DIMENSION B DOES NOT INCLUDE MOLD FLASH
4 DIMENSION L IS TO CENTER OF LEADS WHEN FORMED PARALLEL
5 DIMENSIONING AND TOLERANCING PER ANSI Y14 5, 1982
6 CONTROLLING DIMENSION INCH

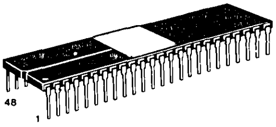
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	61 34	62 10	2 415	2 445
B	13 72	14 22	0 540	0 560
C	3 94	5 08	0 155	0 200
D	0 36	0 55	0 014	0 022
F	1.02	1 52	0 040	0 060
G	2 54 BSC		0 100 BSC	
H	1 79 BSC		0 070 BSC	
J	0 20	0 38	0 008	0 015
K	2 92	3 42	0 115	0 135
L	15 24 BSC		0 600 BSC	
M	0°	15°	0°	15°
N	0 51	1 01	0 020	0 040

CERAMIC PACKAGE
Case 740-03


-A-
48
25
1
24
-B-
N
C
K
L
-T-
SEATING PLANE
E
G
F
M
D 48 PL
J 48 PL
0.25 (0.010) Ⓜ T A Ⓜ
0.25 (0.010) Ⓜ T B Ⓜ

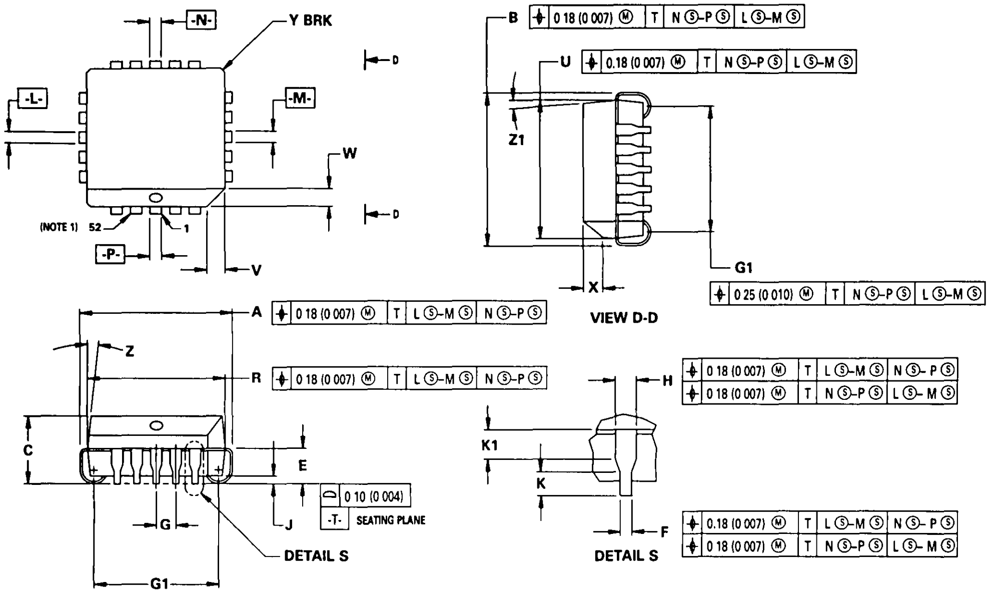
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIM L TO CENTER OF LEAD WHEN FORMED PARALLEL

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	60.36	61.56	2.376	2.424
B	14.64	15.34	0.576	0.604
C	3.05	4.31	0.120	0.170
D	0.381	0.533	0.015	0.021
E	1.27 BSC		0.050 BSC	
F	0.762	1.397	0.030	0.055
G	2.54 BSC		0.100 BSC	
J	0.204	0.330	0.008	0.013
K	2.54	4.19	0.100	0.165
L	15.24 BSC		0.600 BSC	
M	0°	10°	0°	10°
N	1.016	1.524	0.040	0.060



52-PIN PACKAGE

PLCC PACKAGE
Case 778-02


-N-
Y BRK
D
-L-
-M-
W
D
(NOTE 1) 52
1
-P-
V
B | ⌖ | 0.18 (0.007) Ⓜ | T | N Ⓢ-P Ⓢ | L Ⓢ-M Ⓢ
U | ⌖ | 0.18 (0.007) Ⓜ | T | N Ⓢ-P Ⓢ | L Ⓢ-M Ⓢ
Z1
G1
X
⌖ | 0.25 (0.010) Ⓜ | T | N Ⓢ-P Ⓢ | L Ⓢ-M Ⓢ
VIEW D-D
A | ⌖ | 0.18 (0.007) Ⓜ | T | L Ⓢ-M Ⓢ | N Ⓢ-P Ⓢ
Z
R | ⌖ | 0.18 (0.007) Ⓜ | T | L Ⓢ-M Ⓢ | N Ⓢ-P Ⓢ
C
E
0.10 (0.004)
-T- SEATING PLANE
J
G
DETAIL S
G1
⌖ | 0.25 (0.010) Ⓢ | T | L Ⓢ-M Ⓢ | N Ⓢ-P Ⓢ
H
⌖ | 0.18 (0.007) Ⓜ | T | L Ⓢ-M Ⓢ | N Ⓢ-P Ⓢ
⌖ | 0.18 (0.007) Ⓜ | T | N Ⓢ-P Ⓢ | L Ⓢ-M Ⓢ
K1
K
F
⌖ | 0.18 (0.007) Ⓜ | T | L Ⓢ-M Ⓢ | N Ⓢ-P Ⓢ
⌖ | 0.18 (0.007) Ⓜ | T | N Ⓢ-P Ⓢ | L Ⓢ-M Ⓢ
DETAIL S

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	19.94	20.19	0.785	0.795
B	19.94	20.19	0.785	0.795
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	—	0.020	—
K	0.64	—	0.025	—
R	19.05	19.20	0.750	0.756
U	19.05	19.20	0.750	0.756
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	—	0.50	—	0.020
Z	2°	10°	2°	10°
G1	18.04	18.54	0.710	0.730
K1	1.02	—	0.040	—
Z1	2°	10°	2°	10°

NOTES

1. DUE TO SPACE LIMITATION, CASE 778-02 SHALL BE REPRESENTED BY A GENERAL (SMALLER) CASE OUTLINE DRAWING RATHER THAN SHOWING ALL 52 LEADS
2. DATUMS -L-, -M-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT PLASTIC BODY AT MOLD PARTING LINE.
3. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
4. DIM R AND U DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 (0.010) PER SIDE.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982
6. CONTROLLING DIMENSION. INCH

Evaluation Modules

Volume II

EVALUATION MODULES

Motorola has developed a series of inexpensive evaluation modules (EVMs) to support the M6801, M6804, M6805, M68HC11 Families of microcontroller units (MCUs) and the MC68HC99 Hard Disk Controller (HDC).

These EVMs aid in designing, debugging, and evaluating the MCU-based target system. This is accomplished by providing monitor debugging firmware, one-line assembler/disassembler, EPROM/EEPROM MCU programming, and host computer down loading capabilities.

Interfacing is accomplished through an RS-232C compatible terminal/host computer Input/Ouput (I/O) ports and MCU expansion I/O ports. The user need only provide a power supply and RS-232C compatible terminal. Table 5-1 provides a quick reference to the MCUs supported by the EVMs. An evaluation products brochure (BR292 R1) is also available for more information on the EVMs.

The following paragraphs provide additional information application to a specific EVM.

M68701EVM — Dual 64K-Byte Memory Maps; 8K Monitor RAM/ROM, 4K Users Pseudo ROM, MCU (Expanded Multiplexed/Single Chip Mode) Extended I/O Ports. Literature available Brochure BR285/D.

M68HC04EVM — 8K Bytes Monitor EPROM, 4K Bytes Pseudo ROM/RAM, 20-PIN MCU Extension I/O Port HMOS/HCMOS Compatible, 28-Pin Extension I/O Port HMOS/HCMOS Compatible, EPROM MCU Programmer: 28-Pin Dual-in-Line Package. Literature available Brochure BR288/D.

M68705EVM — Dual 4K-Byte Memory Maps; 4K Monitor RAM/ROM, 4K Users Pseudo ROM, EPROM MCU Programmer (28-Pin and 40-Pin DIP Packages), MCU (28-Pin and 40-Pin) Expanded I/O Ports. Literature available Brochure BR291/D.

M1468705EVM — Dual 4K-Byte Memory Maps; 4K Monitor RAM/ROM, 4K User Pseudo ROM, EPROM MCU Programmer 28-/40-Pin DIP and 28-/44-Pin-Lead PLCC (Plastic Leaded Chip Carrier) Packages. Literature available Brochure BR294/D.

M68HC05EVM — Dual 8K-Byte Memory Maps; 6K Monitor EPROM, 8K User Pseudo ROM, Single Chip Mode 40, 52, and 68-Pin MCU Extension I/O Ports-HCMOS Compatible, EEPROM MCU Programmer, EPROM MCU Programmer: 40-Pin Dual-in-Line Packages, 40/52-Lead PLCC (Plastic Leaded Chip Carrier) Packages, 28 Pin DIP. Literature available Brochure BR295/D.

MC68HC05M4EVM — Dual 16K-Byte Memory Map; 8K Monitor EPROM; 16K User Pseudo ROM; 68-Pin (Two 34-Pin Connectors) MCU Extension I/O Port; HCMOS/High-Voltage MCU I/O Port Selection Capabilities. Literature available Brochure BR569/D.

M68HC11EVM — Dual 64K-Byte Memory Map; 8K Byte Monitor EPROM, 16K Byte User Pseudo ROM, Expanded Multiplexed Mode and Single Chip Mode MCU Extension I/O Ports-HCMOS Compatible, EEPROM MCU Programmer; 48-Pin Dual-in-Line Package, 52-Lead PLCC (Plastic Leaded Chip Carrier) Packages. Literature available Brochure BR266/D.

M68HC11EVB — 46K-Byte Memory Map; 8K Monitor EPROM, 8K/16K User ROM, Evaluates Single Chip Mode Only, Replaces MCU I/O Ports B and C for EVB Single Chip Mode Evaluation, 60-Pin MCU extension I/O Port HCMOS Compatible. Literature available Brochure BR278/D.

M68HC99EVM — Dual 64K-Byte Memory Map; 8K Byte Monitor EPROM, 16K Byte User Pseudo ROM, Expanded Multiplexed Mode and Single Chip Mode MCU Extension I/O Ports HCMOS Compatible. Literature available Brochure BR512/D.

Table 5-1. EVM Selector Guide

	Evaluation Modules/Boards								
Device Support	MC68701EVM	MC68705EVM	M68HC04EVM	MC68HC05EVM	M68HC05M4EVM	M68HC11EVM	M68HC99EVM	M1468705EVM	M68HC11EVB
MC6801	●								
MC6801U4	●								
MC68701	●								
MC68701U4	●								
MC6803	●								
MC6803U4	●								
MC6804J1/J2			●						
MC6804P2			●						
MC68704P2			●						
MC68HC04P2/P3			●						
MC6805P2/P4/P6		●							
MC6805R2/R3		●							
MC6805U2/U3		●							
MC68705P3/P5		●							
MC68705R3		●							
MC68705U3/U5		●							
MC146805F2								●	
MC146805G2								●	
MC1468705F2								●	
MC1468705G2								●	
MC68HC05A6				●					
MC68HC05B4/B6				●					
MC68HC05C2/C4/C8				●					
MC68HC05L6				●					
MC68HC705C8				●					
MC68HC805B6				●					
MC68HC805C4				●					
MC68HCL05C4/C8				●					
MC68HSC05C4/C8				●					
MC68HC05M4					●				
MC68HC11A0/A1/A8						●			●
MC68HC11E0/E1/E9						●			●
MC68HC811A2						●			●
MC68HC811E2						●			●
MC68HC99							●		

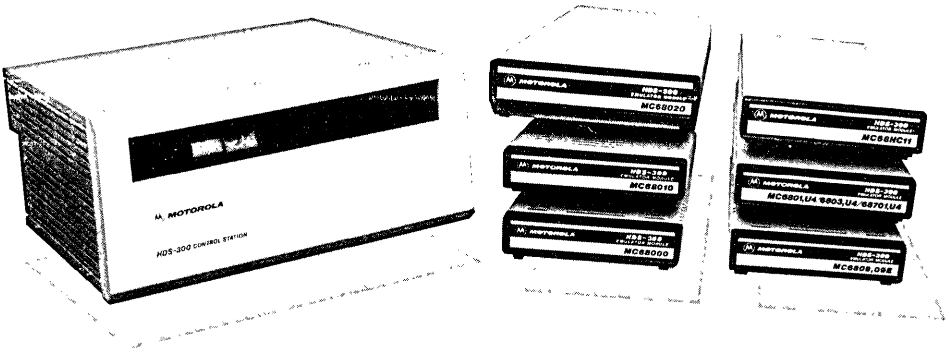
DEVELOPMENT STATION

HDS-300 MICROCONTROLLER HARDWARE/SOFTWARE DEVELOPMENT STATION

The HDS-300 supports the following MCUs and microprocessors (MPUs): MC68HC11A8, MC6801/03, MC68701, MC6809, and MC68HC05C4/C8. The following are some of the features available on the HDS-300.

- Real-Time Emulation, Bus State Monitor, and System Performance Analyzer (SPA)
- Real-Time Trace with Disassembly and 5-1/4 Inch Disk Storage,
- "C" Language Source Level Debug and Formatted Screen Displays,
- One-Line Assembler/Disassembler and Multi-Level Help Display,
- Multiple Station Synchronization and Powerup Self-Test Capability,
- RS-232C Terminal/Host Interface Ports and Centronics Compatible Printer Interface Port.

For further information on the HDS-300 order data sheet HDS-300/DD.



HDS-300 Microcontroller Hardware/Software Development Station

Design Features

- Real-Time Emulation
- Bus State Monitor
- System Performance Analyzer (SPA)
- Real-Time Trace with Disassembly
- "C" Language Source Level Debug
- One-Line Assembler/Disassembler
- Multi-Level Help Display
- Multiple Station Synchronization
- RS-232C Terminal/Host Interface Ports
- Centronics-Compatible Printer Interface Port

The HDS-300 development system serves as the key link between the host system and the target Microcontroller Unit (MCU) or Microprocessor Unit (MPU). The development station provides a quick, user-friendly way to reduce engineering costs and to minimize the risk of failure. The HDS-300 has the capability to operate either as a standalone or with a development host system.

With the host development system, software can be developed on an RS-232C compatible host and then the object code can be downloaded to the HDS-300 for target emulation and debug. When performing source level debug, a hosted HDS-300 displays source and compiled code (including mnemonics) to allow easy modification or step-by-step analysis at either level.

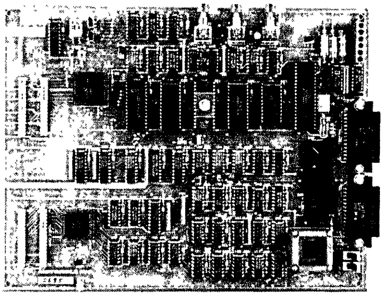
A Bus State Monitor provides trace history as well as real-time trace analysis with disassembly. The use of windows allows easy examination of the trace history of the target system. The HDS-300 allows up to 64 breakpoints to be simultaneously active which streamlines debugging and code verification. Other features include plus user macros, emulation memory, target status analysis, HELP screens, and user-friendly windows.

Emulators

The emulators are used to link the HDS-300 and host system to the target system. The emulators plug directly into the socket that the MCU or MPU would normally occupy and provide the proper electrical connections to duplicate the normal MPU/MCU functions in the target system. Emulators for the HDS-300 System are available to support the M6801, M6803, M6809/E, M68HC05C4/C8 and M68HC11 MCU/MPU Families, as well as the MC68000 Family of 16- and 32-bit MPUs.

Source Level Debugger

The Source Level Debugger (SLD) is a window-based debugger which allows you to view and manipulate the target system via source code. Key features include single stepping, free running execution, restart execution from the beginning of the application, and the ability to set breakpoints at the source line, function, or physical address. This allows you to see how the compiler handles your source code, instruction by instruction, and then allows you to reprogram where necessary. Other features include the displaying of intermixed source and assembly code, and scrolling of source windows.



LOW COST EVALUATION MODULES

A series of inexpensive evaluation modules (EVMs) are available for designing, debugging, and evaluating Motorola microprocessors (MPUs) and microcontrollers (MCUs) in target system equipment. The EVMs provide all of the essential MPU/MCU timing. The on-board ROM contains extensive commands for controlling input/output (I/O) and debug operations, including Motorola S-record transfer.

Memory, internal registers, and I/O registers may be displayed and modified. Program execution may be traced one step at a time or breakpoints may be inserted for program interruption. Circuitry and firmware are included to allow one time programmable read-only memory (OTPROM), erasable programmable read-only memory (EEPROM) MCUs to be programmed. The EVMs require only a power supply and RS-232C compatible terminal for operation.

M68701EVM

The M68701EVM was designed to evaluate the MC6801, MC6801U4, MC6803, MC6803U4, MC68701, and MC68701U4 based target system equipment.

This EVM operates in either the expanded multiplexed mode (mode 2) or the single chip mode (mode 7). Generating, executing, and debugging of target system MCU code can be accomplished in either mode. If mode 2 is used, the target system ROM must be replaced with RAM. This replacement enables operations such as breakpoint, trace, download, assemble, disassemble, and memory modify operations. The host and terminal baud rates are hardware selectable from 150 bit-per-second (bps) to 19.2 Kbps. An EPROM programmer is also provided on the EVM.

M68705EVM and MC1468705EVM

The M68705EVM evaluates the MC6805P2/P6, MC6805R2/R3, MC6805U2/U3, MC68705P3/P5, MC68705R3, and MC68705U3/U5 HMOS MCU-based target system equipment. The M1468705EVM evaluates the MC146805F2 and MC146805G2 CMOS MCU-based target system equipment.

Data transfer within the EVM is controlled by the monitor ROM firmware. User object code may be downloaded into user program RAM (Pseudo ROM) via the host port. The host and terminal port ACIA baud rates are hardware selectable from 100 bps to 19.2 Kbps and both EVMs have EPROM programmers.

M68HC04EVM

The M68HC04EVM evaluates the MC6804J1, MC6804J2, MC6804P2, MC68704P2, MC68HC04P2, and the MC68HC04P3 MCU-based target system equipment. This module contains two resident MCUs, the MC68HC04P3 and the MC6803U4. The MC68HC04P3 is used for evaluating the M6804 Family of HMOS and HCMOS devices and the MC6803U4 controls the EVM monitor functions.

The EVM has an independent hardware baud rate selection of 300 to 19.2 Kbps. Selectable options such as host baud rate, port configuration, IRQ input, EPROM programmer, and clock divide and input selection are provided on the EVM.

M68HC05EVM

The M68HC05EVM evaluates the MC68HC05B5/B6, MC68HC05C2/C3/C4/C8/C9, MC68HC05A6, MC68HC05L6, MC68HC705C8, MC68HC805B6, and MC68HC805C4 HCMOS MCU based target system equipment. The EVM has limitations on evaluating the MC68HCL05C4/C8 and MC68HSC05C4/C8 MCUs in respect to power and speed, respectively.

The terminal port has a 9600 fixed baud rate and the host port has software selectable baud rate of 300 to 19.2 Kbps. Selectable options such as IRQ sensitivity, clock input selection, and OTPROM/EPROM/EEPROM programmer are provided on the EVM.

M68HC05M4EVM

The M68HC05M4EVM evaluates a MC68HC05M4 HCMOS MCU-based target system equipment. The terminal port has a 9600 fixed baud rate and the host port has software selectable baud rate of 300 to 19.2 Kbps. Selectable options such as IRQ sensitivity and clock input selection are provided on the EVM. This EVM has HCMOS/high-voltage MCU I/O port selection capabilities.

M68HC11EVM and M68HC11EVB

The M68HC11EVM evaluates both the M68HC11A8 and M68HC11E9 Family devices.

The EVM has a auto-selectable baud rate for the terminal from 150 to 19.5 Kbps and a software selectable baud rate for the host. Jumper selectable options such as evaluation mode, clock input selection, and EEPROM MCU programmer are provided on the EVM.

The EVB evaluates the MC68HC11A0/A1/A8 only and the MC68HC811A2 with some restrictions. The EVB was designed along with a monitor/debugging program called BUFFALO (Bit User Fast Friendly Aid to Logical Operations) as a low cost alternate to the M68HC11EVM. The EVB only emulates the single chip mode of operation, but operates in the expanded multiplexed mode at all times. Jumper selectable options such as evaluation mode and clock input selection are provided on the EVM. The EVB has no EEPROM programmer.

M68HC99EVM

This module evaluates the MC68HC99 Hard Disk Controller (HDC) based target system equipment. The EVM operates in either the expanded multiplexed (1, 2, or 3) or single chip mode.

The EVM has a auto-selectable baud rate for the terminal from 150 to 19.2 Kbps and a software selectable baud rate from 150 to 19.2 Kbps for the host. Jumper selectable options such as evaluation mode and clock input selection are available.

This chapter contains the ordering forms for the devices in Chapter 3. A copy of the form(s) may be submitted when ordering to maintain integrity of the data book. Please ensure all applicable blanks are filled in.

Electronic transfer of Read Only Memory (ROM) patterns is scheduled to be available by the end of 1988. Contact your local Motorola sales office or representative to find out when this service will be available.

Ordering Information Forms

Volume II

CUT HERE

MC6801/MC6803 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

Mask Option List

Type of MCU ☐ MC6801 ☐ MC6803

Temperature Range
☐ 0° to 70°C
☐ −40° to +85°C

Pattern Media
Diskette: ☐ MDOS Disk File
☐ PC-DOS Disk File
EPROM: ☐ Two 2516 or 2716
☐ Other ____________________
(Requires prior factory approval.)

Package Type
☐ Cerdip
☐ Plastic

Frequency of Operation
☐ 1.0 MHz
☐ 1.25 MHz
☐ 2.0 MHz

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6801U4/MC6803U4 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

Mask Option List

Type of MCU ☐ MC6801U4 ☐ MC6803U4

Temperature Range
- ☐ 0° to 70°C
- ☐ −40° to 85°C

Pattern Media
Diskette:
- ☐ MDOS Disk File
- ☐ PC-DOS Disk File

EPROM:
- ☐ Two 2516 or 2716
- ☐ 2532 or 2732
- ☐ MC68701U4
- ☐ Other ____________________
(Requires prior factory approval.)

Package Type
- ☐ Cerdip
- ☐ Plastic

Frequency of Operation
- ☐ 1.0 MHz
- ☐ 1.25 MHz

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6804J1 ORDERING FORM

Date ______________ Customer PO Number ______________

Customer Company ______________

Address ______________

City ______________ State ______ Zip ______________

Country ______________

Phone ______________ Extension ______________

Customer Contact Person ______________

Customer Part Number ______________

(12 Characters Maximum—If Applicable)

MC6804J1 MASK OPTION LIST

Internal Oscillator Input
☐ Crystal
☐ Resistor

Interrupt Trigger
☐ Edge-Sensitive
☐ Level-Sensitive

Output Drive (Select one Option per Port)

	LSTTL	CMOS/LSTTL	Open Drain
Port A	☐	☐	☐
Port B	☐	☐	☐

Test Clamping Diode
(Available only with LSTTL or Open Drain Outputs)
☐ Yes
☐ No

Temperature
☐ 0°C to 70°C
☐ −40°C to +85°C

Clock Frequency: ______________

Special Electrical Provisions. ______________

(Customer specifications required.)

Pattern Media

Diskette.
☐ MDOS Disk File
☐ MS-DOS Disk File
☐ PC-DOS Disk File

EPROM:
☐ Two 2516 or 2716
☐ One 2532 or 2732
☐ Other ______________

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

MC6804J2 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____
(12 Characters Maximum—If Applicable)

MC6804J2 MASK OPTION LIST

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Level-Sensitive

Output Drive (Select one Option per Port)

	LSTTL	CMOS/LSTTL	Open Drain
Port A	☐	☐	☐
Port B	☐	☐	☐

Test Clamping Diode
(Available only with LSTTL
or Open Drain Outputs)
- ☐ Yes
- ☐ No

Temperature
- ☐ 0°C to 70°C
- ☐ −40°C to +85°C

Clock Frequency: ______________________

Special Electrical Provisions: ____
(Customer specifications required.)

Pattern Media

Diskette: ☐ MDOS Disk File ☐ PC-DOS Disk File
☐ MS-DOS Disk File

EPROM: ☐ Two 2516 or 2716 ☐ One 2532 or 2732
☐ Other ____
(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

CUT HERE

MC6804P2 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____
(12 Characters Maximum—If Applicable)

MC6804P2 MASK OPTION LIST

Internal Oscillator Input
☐ Crystal
☐ Resistor

Interrupt Trigger
☐ Edge-Sensitive
☐ Level-Sensitive

Output Drive (Select one Option per Port)

	LSTTL	CMOS/LSTTL	Open Drain
Port A	☐	☐	☐
Port B	☐	☐	☐
Port C	☐	☐	☐

Test Clamping Diode
(Available only with LSTTL or Open Drain Outputs)
☐ Yes
☐ No

Temperature
☐ 0°C to 70°C
☐ −40°C to +85°C

Clock Frequency: ______________________

Special Electrical Provisions: ____
(Customer specifications required.)

Pattern Media
Diskette: ☐ MDOS Disk File ☐ PC-DOS Disk File
☐ MS-DOS Disk File
EPROM: ☐ Two 2516 or 2716 ☐ One 2532 or 2732
☐ Other ____
(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC04J2 ORDERING FORM

Date ______________ Customer PO Number ______________

Customer Company ______________

Address ______________

City ______________ State ________ Zip ______________

Country ______________

Phone ______________ Extension ______________

Customer Contact Person ______________

Customer Part Number ______________
(12 Characters Maximum—If Applicable)

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor-Capacitor

Clock Predivide Ratio:
- ☐ Divide-by-One
- ☐ Divide-by-Two
- ☐ Divide-by-Four

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level-Sensitive

I/O Options:	Pulldowns	No Pulldowns
PA4, 5, 6, 7	☐	☐
PB0	☐	☐
PB1, 2	☐	☐
PB3, 4, 5, 6, 7	☐	☐

Input Options:	Pullups	No Pullups
Reset	☐	☐
Interrupt	☐	☐

Security:	Protected	No Protection
Program ROM	☐	☐

Voltage Supply (Select any or all): ☐ 2.2 V ± 10% ☐ 3.0 V ± 10% ☐ 5.0 V ± 10%

Temperature Range:

Clock Frequency: ______________

Special Electrical Provisions: ______________
(Customer specifications required.)

Pattern Media

Diskette: ☐ MDOS Disk File ☐ MS-DOS Disk File

EPROM ☐ Two 2516 or 2716 ☐ One 2532 or 2732

☐ Other ______________
(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC04J3 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________

(12 Characters Maximum—If Applicable)

Internal Oscillator Input:	Clock Predivide Ratio:	Interrupt Trigger
☐ Crystal/Resonator	☐ Divide-by-One	☐ Edge-Only Sensitive
☐ Resistor-Capacitor	☐ Divide-by-Two	☐ Edge/Level Sensitive
	☐ Divide-by-Four	

I/O Options:	Pulldowns	No Pulldowns
PA4, 5, 6, 7	☐	☐
PB0	☐	☐
PB1, 2	☐	☐
PB3, 4, 5, 6, 7	☐	☐

Input Options:	Pullups	No Pullups
Reset	☐	☐
Interrupt	☐	☐

Security:	Protected	No Protection
Program ROM	☐	☐

Voltage Supply (Select any or all): ☐ 2.2 V ± 10% ☐ 3.0 V ± 10% ☐ 5.0 V ± 10%

Temperature Range: Clock Frequency: ____________________

Special Electrical Provisions: ____________________

(Customer specifications required.)

Pattern Media

Diskette:	☐ MDOS Disk File	☐ MS-DOS Disk File
EPROM	☐ Two 2516 or 2716	☐ One 2532 or 2732
	☐ Other ____________________	

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805P2 ORDERING FORM

Date ________________ Customer PO Number ________________

Customer Company ________________

Address ________________

City ________________ State ________ Zip ________________

Country ________________

Phone ________________ Extension ________________

Customer Contact Person ________________

Customer Part Number ________________

(12 Characters Maximum—If Applicable)

Timer Clock Source
- ☐ Internal $\phi2$ clock
- ☐ TIMER input pin

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL only

Clock Freq. ________________

Temp. Range ________ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2516 or 2716 EPROM
- ☐ MC68705P3
- ☐ Other ________________

(Requires prior factory approval.)

- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File

________________ (SIGNATURE) Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

________________ (SIGNATURE) Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805P6 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________ State __________ Zip ______________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________
(12 Characters Maximum—If Applicable)

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL only

Clock Freq. ______________

Temp. Range ______________ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2516 or 2716 EPROM
- ☐ MC68705P3
- ☐ Other ______________________
  (Requires prior factory approval.)
- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805R2 ORDER FORM

Date ________________ Customer PO Number ________________

Customer Company ________________

Address ________________

City ________________ State ________ Zip ________________

Country ________________

Phone ________________ Extension ________________

Customer Contact Person ________________

Customer Part Number ________________

(12 Characters Maximum—If Applicable)

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port C Output Drive
- ☐ TTL
- ☐ Open-Drain

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Clock Frequency ________________

Temperature Range ________________ 0°C to +70°C (Standard)
−40°C to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532/2732 EPROM
- ☐ 2516/2716 EPROM
- ☐ Other ________________ (Requires prior factory approval.)
- ☐ PC-DOS Disk File
- ☐ 68705R3 EPROM
- ☐ MS-DOS Disk File

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805R3 ORDERING FORM

Date ______________ Customer PO Number ______________

Customer Company ______________

Address ______________

City ______________ State ______ Zip ______________

Country ______________

Phone ______________ Extension ______________

Customer Contact Person ______________

Customer Part Number ______________
(12 Characters Maximum—If Applicable)

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Port C Output Drive
- ☐ TTL
- ☐ Open-Drain

Clock Frequency ______________

Temperature Range ______________ 0°C to +70°C (Standard)
−40°C to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532/2732 EPROM
- ☐ 2516/2716 EPROM
- ☐ Other ______________ (Requires prior factory approval.)
- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

MC6805S2 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

Standby RAM
- ☐ None
- ☐ 16-byte

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only (CMOS disabled)

Temperature
- ☐ 0°C to 70 C
- ☐ -40 C to 85 C

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Clock Frequency: __________

Low Voltage Inhibit
- ☐ Enabled
- ☐ Disabled

Fifth A D Channel
- ☐ Enabled
- ☐ Disabled

Special Electrical Provisions. ____________________
(Requires prior factory approval.)

____

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532 EPROMs
- ☐ 2716 EPROMs (2)
- ☐ Other ____________________
- ☐ PC-DOS Disk File
- ☐ 2732 EPROMs
- ☐ 2516 EPROMs (2)
- MS-DOS Disk File

(Customer specificatons required.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

CUT HERE

MC6805S3 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

Standby RAM
- ☐ None
- ☐ 16-byte

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Enabled
- ☐ Disabled

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only (CMOS disabled)

Fifth A D Channel
- ☐ Enabled
- ☐ Disabled

Clock Frequency: __________

Temperature
- ☐ 0°C to 70°C
- ☐ −40°C to 85°C

Special Electrical Provisions: ______________________

(Requires prior factory approval.)

____

Pattern Media

- ☐ MDOS Disk File
- ☐ 2532 EPROMs
- ☐ 2716 EPROMs (2)
- ☐ Other ______________________
- ☐ PC-DOS Disk File
- ☐ 2732 EPROMs
- ☐ 2516 EPROMs (2)
- MS-DOS Disk File

(Customer specificatons required)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805U2 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port C Output Drive
- ☐ TTL
- ☐ Open-Drain

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Clock Frequency ______________________

Temperature Range ______________________ 0°C to +70°C (Standard)
−40°C to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532/2732 EPROM
- ☐ 2516/2716 EPROM
- ☐ Other ______________________
- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File
- ☐ MS-DOS Disk File

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC6805U3 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port C Output Drive
- ☐ TTL
- ☐ Open-Drain

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Clock Frequency ______________________

Temperature Range ______________________ 0°C to +70°C (Standard)
−40°C to +85°C

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532/2732 EPROM
- ☐ 2516/2716 EPROM
- ☐ Other ______________________

(Requires prior factory approval.)

- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05B4 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____

(12 Characters Maximum—If Applicable)

MC68HC05B4 MCU ORDERING FORM

Temperature Range:
- ☐ 0° to 70°C (Standard)
- ☐ −40° to 85°C (Extended)
- ☐ −40° to 125° (Automotive

Watchdog Timer
- ☐ Watchdog Timer Automatically Enabled
- ☐ Watchdog Timer Enabled by Software
- ☐ Watchdog Timer Active during WAIT
- ☐ Watchdog Timer Suspended during WAIT

POR Input Pulse Width
- ☐ 4064 Cycle Option
- ☐ 16 Cycle Option

Special Electrical Providers: ______________________________

(Customer specifications required.)

Pattern Media:

- ☐ MC68HC805B6
- ☐ MCM68764 EPROM
- ☐ Other ______________________________
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05B6 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____
(12 Characters Maximum—If Applicable)

MC68HC05B6 MCU ORDERING FORM

Temperature Range:
- ☐ 0° to 70°C (Standard)
- ☐ −40° to 85°C (Extended)
- ☐ −40° to 125° (Automotive)

Watchdog Timer Enable
- ☐ Watchdog Timer Automatically Enabled
- ☐ Watchdog Timer Enabled by Software
- ☐ Watchdog Timer During WAIT
- ☐ Watchdog Timer Active during WAIT
- ☐ Watchdog Timer Suspended during WAIT

Poweron Reset Delay
- ☐ 16 t_{cyc}
- ☐ 4064 t_{cyc}

Special Electrical Provisions: ______________________________
(Customer specifications required.)

Pattern Media:
- ☐ MC68HC805B6
- ☐ MCM68764 EPROM
- ☐ Other ______________________________
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

MC68HC05C2 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________________________

Address ______________________________________

City ______________________ State __________ Zip ______________________

Country ______________________________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________________________

Customer Part Number ______________________________________

(12 Characters Maximum—If Applicable)

MC68HC05C2 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C
- ☐ −40° to 85°C
- ☐ −40° to 105°C
- ☐ −40° to 125°C

Special Electrical Provisions: ______________________________

(Customer specifications required.)

Pattern Media:
- ☐ MC68HC805C4
- ☐ MCM68764 EPROM
- ☐ Other ______________
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

____

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

CUT HERE

MC68HC05C3 MCU ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________

(12 Characters Maximum—If Applicable)

MC68HC05C3 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C
- ☐ −40° to 85°C
- ☐ −40° to 105°C
- ☐ −40° to 125°C

Special Electrical Provisions: ____________________

(Customer specifications required.)

Pattern Media:

- ☐ MC68HC05C3
- ☐ MCM68764 EPROM
- ☐ Other
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

____

(Requires prior factory approval.)

____________________ (SIGNATURE)

Device to be tested to Motorola data sheet specifications Customer part number, if used as part of marking, is for reference purposes only

____________________ (SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC805C4 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State ____________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________
(12 Characters Maximum—If Applicable)

MC68HC805C4 MCU ORDERING FORM

Internal Oscillator Input:
☐ Crystal/Resonator
☐ Resistor

Temperature Range:
☐ 0° to 70°C (Standard)
☐ −40° to +85°C (Extended)

Special Electrical Provisions: ______________________
(Customer specifications required.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HCL05C8 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____
(12 Characters Maximum—If Applicable)

MC68HCL05C8 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C
- ☐ 25°C (Only)

Special Electrical Provisions: ______________________________
(Customer specifications required.)

Pattern Media:

- ☐ MC68HCL05C8
- ☐ MCM68764 EPROM
- ☐ Other ______________________________
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

(Requires prior factory approval.)

______________________ (SIGNATURE)	Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only
______________________ (SIGNATURE)	Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05L6 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

M68HC05L6 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Temperature Range:
- ☐ 0° to 70°C

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Tone Generator
- ☐ A Crystal Frequency (1024.4)
- ☐ B Crystal Frequency (512.4)
- ☐ C Crystal Frequency (256.4)
- ☐ D Crystal Frequency (64.4)

Special Electrical Providions: ______________________

(Customer specifications required.)

Pattern Media:
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ Other ______________________
- ☐ PC-DOS Disk File

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05P1 MCU ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

MC68HC05P1 MCU ORDERING FORM

Internal Oscillator Input:
☐ Crystal/Resonator
☐ Resistor

Interrupt Trigger
☐ Edge-Sensitive
☐ Edge/Level Sensitive

Temperature Range:
☐ 0° to 70°C (Standard)

Special Electrical Provisions: ____________________
(Customer specifications required.)

Pattern Media:

☐ MC68HC805C4
☐ MCM68764 EPROM
☐ Other ____________________
☐ MS-DOS Disk File
☐ MCM68766 EPROM
☐ PC-DOS Disk File
☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05C4 MCU ORDERING FORM

Date ________________ Customer PO Number ________________

Customer Company ________________

Address ________________

City ________________ State ________ Zip ________________

Country ________________

Phone ________________ Extension ________________

Customer Contact Person ________________

Customer Part Number ________________
(12 Characters Maximum—If Applicable)

MC68HC05C4 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C (Standard)
- ☐ −40° to 85°C (Extended)
- ☐ −40° to 125° (Automotive

Special Electrical Provisions: ________________
(Customer specifications required.)

Pattern Media:

- ☐ MC68HC05C4
- ☐ MCM68764 EPROM
- ☐ Other ________________
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC05C8 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

MC68HC05C8 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C
- ☐ −40° to 85°C
- ☐ −40° to 105°C
- ☐ −40° to 125°C

Special Electrical Provisions: ______________________

(Customer specifications required.)

Pattern Media:
- ☐ MC68HC05C8
- ☐ MCM68764 EPROM
- ☐ Other
- ☐ MS-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

____

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HSC05C4 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________
(12 Characters Maximum—If Applicable)

MC68HSC05C4 MCU ORDERING FORM

Internal Oscillator Input:
- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Edge/Level Sensitive

Temperature Range:
- ☐ 0° to 70°C (Standard)

Special Electrical Provisions: ______________________
(Customer specifications required.)

Pattern Media:
- ☐ MCM68764 EPROM
- ☐ MS-DOS Disk File
- ☐ Other ______________________
- ☐ MCM68766 EPROM
- ☐ PC-DOS Disk File
- ☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HSC05C8 MCU ORDERING FORM

Date ______________ Customer PO Number ______________

Customer Company ______________

Address ______________

City ______________ State ______ Zip ______________

Country ______________

Phone ______________ Extension ______________

Customer Contact Person ______________

Customer Part Number ______________

(12 Characters Maximum—If Applicable)

MC68HSC05C8 MCU ORDERING FORM

Internal Oscillator Input:
☐ Crystal/Resonator
☐ Resistor

Interrupt Trigger
☐ Edge-Sensitive
☐ Edge/Level Sensitive

Temperature Range:
☐ 0° to 70°C (Standard)

Special Electrical Provisions: ______________
(Customer specifications required.)

Pattern Media:
☐ MCM68764 EPROM
☐ MS-DOS Disk File
☐ Other ______________
☐ MCM68766 EPROM
☐ PC-DOS Disk File
☐ 2764 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC705C8 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________
(12 Characters Maximum—If Applicable)

Internal Oscillator Input:
☐ Crystal/Resonator
☐ Resistor

Temperature Range:
☐ 0° to 70°C (Standard)
☐ −40° to 85°C

Special Electrical Provisions: ______________________
(Customer specifications required.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HCL05C4 MCU ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ____

Address ____

City ______________________ State __________ Zip ______________

Country ____

Phone ______________________ Extension ______________________

Customer Contact Person ____

Customer Part Number ____
(12 Characters Maximum—If Applicable)

Internal Oscillator Input:
☐ Crystal/Resonator
☐ Resistor

Temperature Range:
☐ 0° to 70°C
☐ 25°C Only

Interrupt Trigger:
☐ Edge-Sensitive
☐ Edge/Level Sensitive

Special Electrical Provisions: ____
(Customer specifications required.)

Pattern Media:
☐ MCM68764 EPROM
☐ MC68HC805C4
☐ Other ______________________________
☐ MS-DOS Disk File
☐ MCM68766 EPROM
☐ PC-DOS Disk File
☐ 2764 EPROM
(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC146805F2 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

Internal Oscillator Input:
- ☐ Crystal
- ☐ Resistor (Not available on B54F mask)

Clock Predivide Ratio:
- ☐ −4
- ☐ −2

Interrupt Trigger:
- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Temperature Range __________

Pattern Media:
- ☐ MDOS Disk File
- ☐ 1468705F2 EPROM
- ☐ 2516 or 2716 EPROM
- ☐ PC-DOS Disk File
- ☐ MS-DOS Disk File
- ☐ Other ____________________
(Requires prior factory approval)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC146805G2 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________ State __________ Zip ______________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

Internal Oscillator Input
☐ Crystal

Internal Divide
☐ ÷4
☐ −2

Interrupt Trigger
☐ Edge-Sensitive Only
☐ Level-Sensitive and Edge-Sensitive

TEMPERATURE ______________________ FREQUENCY ______________________

Pattern Media

☐ MDOS Disk File
☐ 2532 EPROM
☐ 2716/2516(2) EPROM
☐ Other ______________________

☐ PC-DOS Disk File
☐ 2732 EPROM

☐ MS-DOS Disk File
☐ 1468705G2 EPROM

(Requires prior factory approval.)

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC11A8 ORDERING FORM

Date ____________________ Customer PO Number ____________________

Customer Company ____________________

Address ____________________

City ____________________ State __________ Zip ____________________

Country ____________________

Phone ____________________ Extension ____________________

Customer Contact Person ____________________

Customer Part Number ____________________
(12 Characters Maximum—If Applicable)

Temperature Range
- ☐ −40 to +85°C
- ☐ −40 to +105°C
- ☐ −40 to +125°C

Package Type
- ☐ 48-Pin Dual-In-Line Plastic
- ☐ 52-Lead PLCC

On-Chip Functions Used
(Check Functions — Factory Use Only)
- ☐ ROM
- ☐ EEPROM
- ☐ A/D
- ☐ COP
- ☐ SPI
- ☐ SCI
- ☐ Timer

Operating (E Clock) Frequency: ____________________

Security Feature Enabled: Yes ________ No ________

CONFIGuration Register Contents: Hex ($) ____________________

Device Marking:

☐ Motorola Standard
Motorola Logo
Motorola Part Number
Mask and Datecode

☐ Standard with Customer Part Number
Motorola Logo
Motorola Part Number
Customer Part Number
Mask and Datecode

☐ Other ____________________

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC11E9 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________
(12 Characters Maximum—If Applicable)

Temperature Range
- ☐ −40 to +85°C
- ☐ −40 to +105°C (See Note)
- ☐ −40 to +125°C (See Note)

Package Type
- ☐ 52-Lead PLCC

On-Chip Functions Used
(Check Functions — Factory Use Only)
- ☐ ROM
- ☐ EEPROM
- ☐ A/D
- ☐ COP
- ☐ SPI
- ☐ SCI
- ☐ Timer

Operating (E Clock) Frequency:

Security Feature Enabled: Yes ________ No ________

CONFIGuration Register Contents: Hex ($)______________

(The CONFIG register will be programmed if desired; however, the NOSEC bit cannot be cleared on parts with security enabled.)

Device Marking:

☐ Motorola Standard
Motorola Logo
Motorola Part Number
Mask and Datecode

☐ Standard with Customer Part Number
Motorola Logo
Motorola Part Number
Customer Part Number
Mask and Datecode

☐ Other______________
____
____

Note: These high temperature ranges require EEPROM consideration. Please refer to *EEPROM CHARACTERISTICS* before submitting ROM code.

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

CUT HERE

MC68HC811E2 ORDERING FORM

Date ______________________ Customer PO Number ______________________

Customer Company ______________________

Address ______________________

City ______________________ State __________ Zip ______________________

Country ______________________

Phone ______________________ Extension ______________________

Customer Contact Person ______________________

Customer Part Number ______________________

(12 Characters Maximum—If Applicable)

Temperature Range
- ☐ −40 to +85°C
- ☐ −40 to +105°C (See Note)
- ☐ −40 to +125°C (See Note)

Package Type
- ☐ 52-Lead PLCC

On-Chip Functions Used
(Check Functions — Factory Use Only)
- ☐ EEPROM
- ☐ A/D
- ☐ COP
- ☐ SPI
- ☐ SCI
- ☐ Timer

Operating (E Clock) Frequency:

Security Feature Enabled: Yes ________ No ________

CONFIGuration Register Contents: Hex ($)______________

(The CONFIG register will be programmed if desired; however, the NOSEC bit cannot be cleared on parts with security enabled.)

Device Marking:

☐ Motorola Standard
Motorola Logo
Motorola Part Number
Mask and Datecode

☐ Standard with Customer Part Number
Motorola Logo
Motorola Part Number
Customer Part Number
Mask and Datecode

☐ Other______________________
____
____

Note: These high temperature ranges require EEPROM consideration. Please refer to *EEPROM CHARACTERISTICS* before submitting ROM code.

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only

____
(SIGNATURE)

Device to be tested to customer specifications (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

Motorola's Microcontroller and Microprocessor Families
Volume I

1

Reliability
Volume I

2

Data Sheets
Volume I and II

3

Mechanical Data
Volume II

4

Evaluation Modules
Volume II

5

Ordering Information Forms
Volume II

6

1 Motorola's Microcontroller and Microprocessor Families
Volume I

2 Reliability
Volume I

3 Data Sheets
Volume I and II

4 Mechanical Data
Volume II

5 Evaluation Modules
Volume II

6 Ordering Information Forms
Volume II

MOTOROLA

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong.

TR48503B022

MOTOROLA

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong.

TB46503B022